DOCUMENT RESUME

ED 401 399 CE 072 887

AUTHOR Lundgren, Carol A.; And Others
TITLE Teaching Computer Applications.
INSTITUTION Delta Pi Epsilon Society, Little Rock, AR.
REPORT NO ISBN-1-881530-01-9
PUB DATE 95
NOTE 72p.
AVAILABLE FROM Delta Pi Epsilon, Inc., P.O. Box 4340, Little Rock, AR 72214 ($12).
PUB TYPE Guides - Classroom Use - Teaching Guides (For Teacher) (052) -- Reports - Descriptive (141)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Business Education; *Classroom Techniques; *Computer Literacy; *Computer Oriented Programs; Computer Software; Curriculum Development; Database Management Systems; Databases; Desktop Publishing; Educational Technology; Postsecondary Education; Secondary Education; *Spreadsheets; Teaching Methods; *Word Processing

ABSTRACT

This document, which is designed to provide classroom teachers at all levels with practical ideas for a computer applications course, examines curricular considerations, teaching strategies, delivery techniques, and assessment methods applicable to a course focusing on applications of computers in business. The guide is divided into three sections. Section 1 presents an overview of teaching computer applications. Discussed in section 2 are the following topics related to teaching spreadsheet and database software applications: curriculum considerations; evolution of spreadsheet software, evolution of database software; teaching materials and software; technology for teaching; teaching environments, strategies, and evaluation; and evaluation of performance. Section 3 covers the following topics within the context of teaching word processing and desktop publishing: technological influences and the curriculum; teaching strategies and delivery; delivery techniques; and testing and evaluation. Contains 24 figures and 19 references. (MN)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 401 399

TEACHING COMPUTER APPLICATIONS

Carol A. Lundgren
Terry D. Lundgren
George A. Mundrake

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

✓ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL IN OTHER THAN PAPER COPY HAS BEEN GRANTED BY

R B Mitchell

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Delta Pi Epsilon

CE 072 887

BEST COPY AVAILABLE

Delta Pi Epsilon has as part of its program of service publications for researchers and classroom teachers. The publications are written by recognized authorities in the field. *Teaching Computer Applications* is one of the publications designed for immediate application in enlivening and enriching the teaching of business subjects.

The following publications are available for use in classes such as methods of teaching, principles of business education, and research as well as for use by teachers who want to remain abreast of the latest teaching/learning strategies and research techniques:

Design and Conduct of Educational Surveys and Experiments by Leonard J. West
Effective Evaluation in Business Education by B.W. Stehr
Research Methods in Business Education by Padmakar M. Sapre
Research: Process and Product by Mildred Hillestad
Teaching Business Communications by Max L. Waters and Don Leonard
Teaching Consumer Education by David Graf
Teaching General Business by Betty Jean Brown
Teaching Keyboarding by Gary N. McLean
Teaching Computer Applications by Carol A. Lundgren, Terry D. Lundgren, and George A. Mundrake

Information concerning these publications may be obtained from the Delta Pi Epsilon National Office. © Copyright, 1995, by Delta Pi Epsilon. Printed in the United States of America.

1994-1995 DPE NATIONAL EXECUTIVE BOARD
President: Sharon Lund O'Neil; University of Houston; Houston, TX 72204
Vice President: B. June Schmidt; Virginia Polytechnic Institute and State Un.; Blacksburg, VA 24061
Secretary: F. Stanford Wayne; Grandview, OH 43212
Treasurer: Peter F. Meggison; Massasoit Community College; Brockton, MA 02402
Historian: Terry D. Roach; Arkansas State University; State University, AR 72467
Past President: Betty J. Brown; Ball State University; Muncie, IN 47306

1994-1995 DPE NATIONAL PUBLICATIONS COMMITTEE
Chairperson: Clora Mae Baker; Southern Illinois University; Carbondale, IL 62901
Don Bright; Bowling Green State University; Bowling Green, OH 43403
Marcia Bush; CDE, Business Education Unit; Pleasant Hill, CA 94523
Janet Fong; New York University; New York, NY 10012
Randy L. Joyner; East Carolina University; Greenville, NC 27858
Rebecca Limback; Central Missouri State University; Warrensburg, MO 54093
Roger Luft; Eastern Illinois University; Charleston, IL 61920
Beryl McEwen; Eastern Illinois University; Charleston, IL 61920
Jan Patton; UT Satellite Station; Commerce, TX 75428
Heidi Perreault; Southwest Missouri State University; Springfield, MO 65804
Donna H. Redmann; Louisiana State University; Baton Rouge, LA 70803
Betty Rogers; University of Central Arkansas; Conway, AR 72035

DPE NATIONAL OFFICE
Executive Director: Robert B. Mitchell; P.O. Box 4340; Little Rock, AR 72214

TEACHING COMPUTER APPLICATIONS

Carol A. Lundgren
Terry D. Lundgren
George A. Mundrake

DELTA PI EPSILON

TABLE OF CONTENTS

TEACHING COMPUTER APPLICATIONS 1

TEACHING SPREADSHEET AND DATABASE SOFTWARE APPLICATIONS 3

INTRODUCTION 3

CURRICULUM CONSIDERATIONS 3

EVOLUTION OF SPREADSHEET SOFTWARE 4

- Development of Spreadsheet Software 4
- Trends in Spreadsheet Software Design 5

EVOLUTION OF DATABASE SOFTWARE 7

- Purpose and Types of Databases 7
- Trends in Database Management Systems 11

TEACHING MATERIALS AND SOFTWARE 11

- Printed Materials 14
- Software 14
- Site License Agreements 16
- Public Domain Software and Shareware 16

TECHNOLOGY FOR TEACHING 17

TEACHING ENVIRONMENTS, STRATEGIES, AND EVALUATION 17

- Teaching Environments 17
- Teaching in the Computer Lab 19
- Teaching in the Classroom without Computers 19
- Classroom/Laboratory Combinations 20
- Problem Solving 24
- Demonstration 25
- Simulations 26
- Self-Paced Learning Modules 29

EVALUATION OF PERFORMANCE 29

Pretesting and Observation 29
Homework and In-Class Assignments 30
Objective and Performance Tests 31

SUMMARY 33

SPREADSHEET AND DATABASE SOFTWARE REFERENCES 34

TEACHING WORD PROCESSING AND DESKTOP PUBLISHING 35

INTRODUCTION 35

TECHNOLOGICAL INFLUENCES AND THE CURRICULUM 36

Technological Advancements 36
A Word Processing and DTP Curriculum 38

TEACHING STRATEGIES AND DELIVERY 39

Assessing the Learner 39
Establishing Objectives and Content 41

DELIVERY TECHNIQUES 42

Lecture and Demonstration 42
Self-Paced Materials 45
Problem-Solving Learning Activities 45
Simulations 56

TESTING AND EVALUATION FOR WORD PROCESSING AND DESKTOP PUBLISHING 58

Evaluation and Curriculum 58
Variety, Weights, and Frequency in Evaluation Mix 59
Theory Tests 59
Performance Tests 61
Other Evaluation Techniques 64

SUMMARY 64

WORD PROCESSING AND DESKTOP PUBLISHING REFERENCES 66

LIST OF FIGURES

SPREADSHEET AND DATABASE APPLICATIONS

Figure

1 Software Rating Form 6
2 Two- and Three-Dimensional Spreadsheets 8
3 Basic Structure of Data 10
4 Types of Information Processing at Different Levels of Management 12
5 Information Considerations at Different Levels of Management 13
6 Textbook Selection Form 15
7 Possible Projected Image 18
8 Database Protocol Exercise 21
9 Spreadsheet Cell Entry Exercise 22
10 Lotus Spreadsheet Problem with Specified Outcomes 23
11 Problem-Solving Approach in Spreadsheet Problem 27
12 Problem-Solving Approach in Database Problem 28
13 Objective Test Items 32

TEACHING WORD PROCESSING AND DESKTOP PUBLISHING

Figure

1 Advance Organizer 44
2 Sample Word Processing Problem 47
3 Sample Boilerplate Document 48
4 Sample Two-Column Story 50
5 Use of Icons or Graphics for Matching Exercise 51
6 Sample Portfolio for Desktop Publishing Terms 53
7 DTP Exercise 55
8 Take-Home or "Carry-Out" Exam Project 57
9 Sample of Tool Box Matching 62
10 Sample of Graphic Matching 63
11 Sample Group Project Guidelines 65

TEACHING COMPUTER APPLICATIONS

Teaching computer applications presents unique and exciting challenges and opportunities for both teachers and students. If applications courses are to be effective, they must be carefully planned in terms of content and materials, instructional methodology, time allotments, and appropriate assessment techniques.

The employment community is the final judge of the effectiveness of computer applications courses: Are students prepared for the entry-level positions available to them when they enter the workforce? Are they promotable? Can students build on their computer skills so they can perform successfully in postsecondary and college/university courses? Does the curriculum provide courses in a sequence which allows them to develop computer skills in a logical order? How does what students learn in computer applications courses complement other courses in which they are enrolled?

Because technology changes so fast, students must be prepared for lifelong learning. Do they have the problem-solving and decision-making skills they will need to adapt to new software packages and evolving technology? These are among the questions teachers should answer as they decide what to teach in their computer applications courses.

This publication is intended to provide the classroom teacher at all levels with practical ideas for a computer applications course--curricular considerations, teaching strategies, delivery techniques, and assessment methods.

TEACHING SPREADSHEET AND DATABASE SOFTWARE APPLICATIONS

by
Carol A. Lundgren
Terry D. Lundgren

INTRODUCTION

Software sales today are dominated by word processing, spreadsheet, and database packages, respectively. With the growing popularity of spreadsheet and database software, business graduates must have some knowledge of these packages and how to use them. Both software and hardware have become sophisticated, and the business applications for spreadsheet and database software are increasing.

Technology is changing too fast for the average school system to keep up; thus having reasonably up-to-date equipment and software continues to be a critical concern. In all computing environments, teachers must constantly monitor the latest developments and devise methods for implementing them into daily lessons. The purpose of this section is to improve teaching and learning by providing ideas for spreadsheet and database instruction.

CURRICULUM CONSIDERATIONS

The requirements of employers influence what courses are taught and what topics are emphasized in secondary, postsecondary, and vocational curricula. Tech Prep is a program that addresses the needs of business. The Tech Prep program is federally funded through the Carl D. Perkins Vocational and Applied Technology Education Act of 1990. Its purpose is to promote cooperation between business and academia with the goal of producing a workforce that can meet the challenges of our rapidly changing technology and world society (Illinois State Board of Education, 1991). Under the auspices of Tech Prep, the business community, secondary schools, postsecondary schools, and colleges have joined forces to provide students with the skills and competencies they will need to compete in the workforce at entry level and throughout their careers. Database and spreadsheet courses are a necessary part of the curriculum that prepares students for the modern workplace.

The possibilities for teaching spreadsheet and database software applications include 1) teaching both along with other computer applications, which is a common approach in an introductory course; 2) teaching spreadsheet and database software applications in separate courses at increasingly higher levels, and 3) providing students with learning modules on an open-entry/open-exit basis.

Students are expected to become proficient in the use of spreadsheet or database software. Introductory, intermediate, and advanced courses over three semesters would be appropriate. Modules that emphasize particular competencies may be used in programs that are not necessarily organized into semester-long courses. Due to rapidly changing technology, teachers will continually need to create a strong bond with businesses to design appropriate curricula.

The development of curricula must be a joint effort of business teachers, teachers from other disciplines, business personnel, and students. Each business program needs an advisory board composed of business people and educators who review and evaluate curriculum. The advisory board is an excellent source of information for structuring database and spreadsheet content. Feedback from business firms who have employed students and graduates can provide valuable information for curriculum. Business people want students to be proficient in using software, but also to be skilled communicators. Therefore, communication skills (reading, writing, speaking, and listening) are a necessary part of computer applications courses at all levels. Also, personal development skills such as self-discipline, interpersonal relationships, decision making, critical thinking, leadership, and career planning are important.

EVOLUTION OF SPREADSHEET SOFTWARE

Spreadsheets were one of the first business applications for microcomputers. Although the spreadsheet concept had existed for decades with clerks entering numeric data in columns and rows by pencil, the advent of the computer brought electronic spreadsheets, enabling the user to record and manipulate data simultaneously. Students in methods classes as well as student teachers need to know the history of spreadsheet software development, the purpose of spreadsheets and their significance for accounting and introduction to business courses, and trends in spreadsheet software design.

Development of Spreadsheet Software

The history of the development of spreadsheet software is rather murky due to the simultaneous development of microcomputers and the fact that early software was developed by amateurs (Cringley, 1992, p. 68).

1979 VisiCalc was developed by Software Arts and marketed by Personal Software. VisiCalc was designed to run on the 48K Apple II Computer and enjoyed enormous popularity.

1981 SuperCalc was introduced. SuperCalc by Computer Associates was designed to run on the CP/M operating system and incorporated many of the same features as VisiCalc.

1981 Multiplan, developed by Microsoft, Inc., with full-word menus and windows, was introduced.

1983 Lotus 1-2-3 was introduced and became the best-selling microcomputer software product in history (Cringley, 1992, p. 149). Lotus Development Corporation's product was designed for the IBM PC. It included a larger spreadsheet size than other programs provided, as well as database and graphics functions.

Until 1989, Lotus was the most popular spreadsheet package used in businesses and taught in colleges (Lomo-David, 1989). However, after the introduction in 1985 of Windows by Microsoft, Inc., spreadsheet programs capable of running under Windows began to appear and eventually usurped Lotus' position. Windows software acts as a transparent interface between the operating system (DOS) and the application program. Today the major spreadsheet producers are Borland (Quattro Pro), Microsoft (Excel), Lotus Development (Lotus 1-2-3), and Computer Associates (SuperCalc). Both Quattro Pro and Excel were designed specifically for Windows.

Spreadsheets, though standardized in format and operation, have a wide array of functions such as graphical displays of data, statistical analysis, and the ability to automate some operations with specialized user-written programs called macros. However, the more features a spreadsheet program has, the more complex it can become. It can become so difficult to use that it is not suitable for some classes. Teachers may find that a rating sheet, such as the one shown in Figure 1, is helpful for choosing a spreadsheet program for use in the classroom.

Trends in Spreadsheet Software Design

One trend in software is group access, using either groupware or networking. Groupware spreadsheet packages are designed to allow each group member to work on his or her own version of a spreadsheet which can be combined with other group members' spreadsheets into a master spreadsheet. The global assumptions are kept in the master spreadsheet, and the individual spreadsheets are derived from it. The other aspect of a groupware spreadsheet that makes exchanging information easy is that it allows multiple users to access the same spreadsheet. Alternatively, with networking, spreadsheets are kept on the network server where all stations can access them. This approach requires a special version of the

Figure 1

Software Rating Form

SOFTWARE RATING FORM

Good				Poor	
5	4	3	2	1	Menus are easy to read.
5	4	3	2	1	Menus are easy to understand
5	4	3	2	1	Icons are useful representations.
5	4	3	2	1	Dialog boxes (if running under Windows) are easy to use.
5	4	3	2	1	Mouse clicking and dragging procedures are clear.
5	4	3	2	1	Keystroke commands are easy or can be replaced by mouse.
5	4	3	2	1	Frequently used commands are easy to perform.
5	4	3	2	1	Help functions are available throughout the program.
5	4	3	2	1	Help function(s) provide understandable information.
5	4	3	2	1	Software contains easy-to-use tutorials.
5	4	3	2	1	Screen displays are reasonable with expected outcomes. (WYSIWYG)
5	4	3	2	1	Chaining of commands (macros) is easily automated.
5	4	3	2	1	Software builds on previously learned skills.
5	4	3	2	1	Special features are useful for skill level being taught.

Other Features and Comments:

spreadsheet program that allows everyone to view the spreadsheet but has the capability to lock and protect spreadsheets so that only one person can make editing changes at a time.

A second trend in software design is the addition of such sophisticated features as three dimensions, in which data are represented by rows, columns, and ranks. The 3-D model is designed as if there were layers of data. Figure 2 illustrates two- and three-dimension spreadsheets.

A third trend is the move to the GUI (Graphical User Interface) spreadsheet. A major advantage of using a GUI spreadsheet is the ability to multitask, i.e., to perform separate tasks simultaneously. This allows the user to pause while working on a spreadsheet and switch to a word processing or a database management program. The ability to multitask can increase productivity enormously by allowing the user to exchange information between applications, such as importing data from a spreadsheet to a table created with word processing software.

Another advantage is Object Linking and Embedding (OLE) which allows an object such as a graphic created with a different program to be embedded in a spreadsheet. When editing the embedded graphic, the creating program is automatically called, the editing performed, and then the user is returned to the spreadsheet with the new version of the object. GUI is a feature of operating systems, such as Macintosh's System 7 and IBM's OS/2, and operating environment software, such as Windows. Using GUI changes the way the user interacts with the software. For example, in a DOS version of spreadsheet software, operations are accomplished by entering commands and pressing keys, while in the Windows (GUI) version the operator uses the mouse to click, double-click, and drag.

Spreadsheet software typically includes graphing and charting capabilities. With the spreadsheet package, charts can be developed to illustrate numeric data. These capabilities make spreadsheet software more than just a "number crunching" process.

EVOLUTION OF DATABASE SOFTWARE

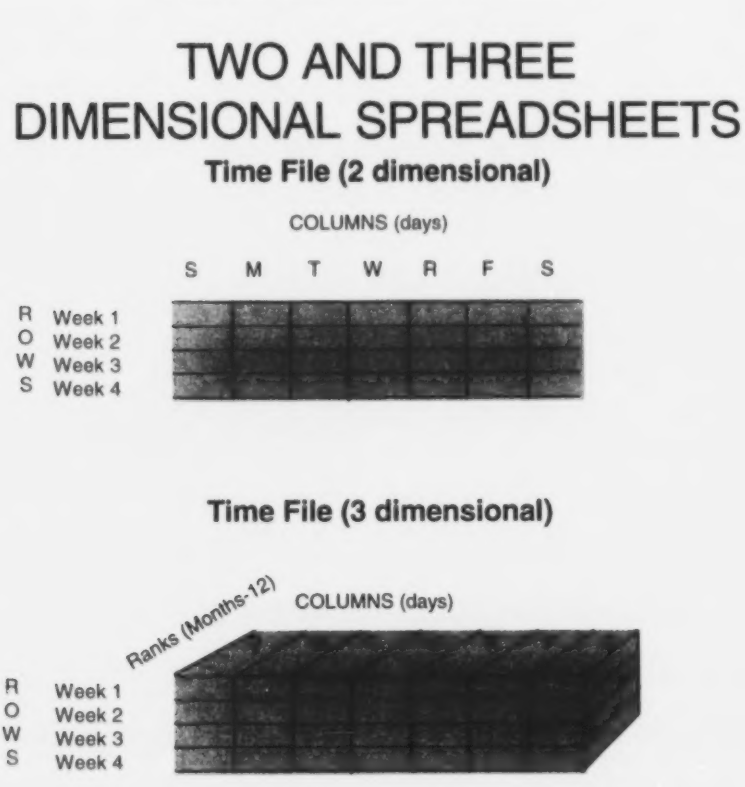
Microcomputer database application systems allow a business to organize and control its data. The business determines what data are appropriate and how data retrieval should take place. Then a database is designed to accommodate those needs. Database software is used to accomplish these objectives.

Purpose and Types of Databases

The purpose of a database is to provide a common depository for data and a means to organize and report that data so that it becomes useful information. A

Figure 2

Two and Three Dimensional Spreadsheets



telephone directory is a database of telephone numbers and, to some extent, addresses. A library card catalog is a database of authors, book titles, and subjects. The organization of both of these databases enables the user to extract needed information. A telephone book organized by phone number would not be very useful to the general population; therefore, the intended use determines the organization of the elements of a database. A database is purposeful to the extent that its elements are coherent (related to a specific activity or purpose) and organized appropriately.

The concepts of coherence and organization are essential to understanding database software applications. Students have had many experiences with databases in everyday life before they use a computer database. That background knowledge can be brought to the task of understanding database concepts before getting down to the details of learning the software. Figure 3 illustrates the basic structure of data that students must understand. Each database software package has its own formats, but the database created with one software package may be used in another with some manipulation. For economic reasons, schools usually invest in only one brand of database software, and students learn only one, often with the result that they are unaware that others exist. Teaching software per se should not be the goal of the computer applications curriculum, but using various database software packages enables students to see the similarities. A thorough knowledge of each software package is not necessary if students understand the basic concepts of database design and management.

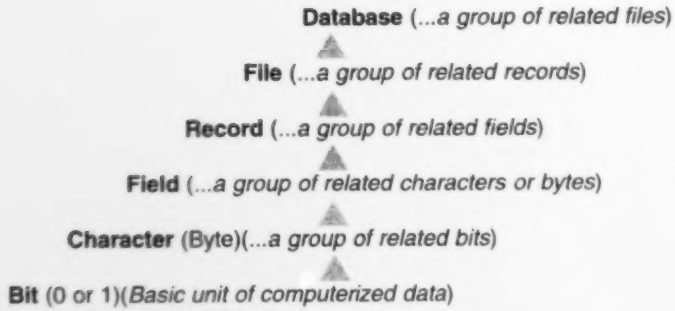
Database software was designed solely to store millions of bits of data. Now database software packages perform many more functions. The earliest database software was designed to store information so that it could be retrieved and manipulated rapidly and in some structured format. Browsing and editing the created database was also possible. Eventually more user-friendly features were added, such as form designers for constructing data entry screens and report builders for generating customized reports. User demand led to the addition of programming in a database management language (DML). When Ashton-Tate introduced dBase, it set the standard for DOS-based database management programs, and dBase dominated the market for many years. As any company with a popular product, Ashton-Tate soon had competitors with database software that was easier to use and more sophisticated. Today, popular packages include dBase IV and Paradox (Borland International), R:Base (Microrim), DataEase (DataEase International), Clipper (Computer Associates), and FoxPro and Access (Microsoft Corp.) Clearly, the software field is dynamic. Companies merge, acquire products, and diversify. Through it all, the consumers (many of whom are educators) must try to make informed choices.

Database management has been viewed largely as a technological function; but in teaching database concepts, teachers should consider two perspectives, one

Figure 3

Basic Structure of Data

The Structure of Database


Database (...a group of related files)
File (...a group of related records)
Record (...a group of related fields)
Field (...a group of related characters or bytes)
Character (Byte)(...a group of related bits)
Bit (0 or 1)(Basic unit of computerized data)

of which emerged from data processing and management information systems traditions and the other from a human resources background (Rathswohl, 1990). Both approaches to information management are necessary, and both perspectives are legitimate topics in computer applications courses.

The data processing perspective of information resource management is that a database management system (DBMS) is a means to manage large amounts of business data. It stresses that the minimal functions of a DBMS are to store and efficiently retrieve business data, and that retrieval ranges from finding a particular piece of information to the production of regular summary reports on the database. Figure 4 illustrates different processing systems for levels of management.

The human resources perspective of information resource management is that the value of the data resides primarily in the people who use it. It emphasizes the fact that database software does not teach what information is needed nor how to use data to create the necessary information. Figure 5 illustrates information needs of different levels of management. Teachers should explain both perspectives when the topic of database management is introduced.

Trends in Database Management Systems

The future of database management applications appears to be associated with trends toward decentralization and end-user computing. Separate databases for distinct operations within a business may be linked to communications software. The development of more powerful computer hardware and increasingly more efficient software has made it possible for each department in a business to have its own database for distinctive and unique operations. Obviously, when such a database is developed, little thought is given to compatibility with other hardware or software; therefore, a major disadvantage of separate databases is lack of access by other departments. Consequently, an associated trend is to link separate databases using gateway software that allows different databases to link and to relate to each other. Gateway software allows the user to access all systems using standard commands and menus.

TEACHING MATERIALS AND SOFTWARE

To teach spreadsheet and database software, a teacher must carefully select appropriate materials and software. With the abundance of materials and software available, that selection is a difficult task. Complicating the task is rapid technological change.

Figure 4

Types of Information Processing at Different Levels of Management

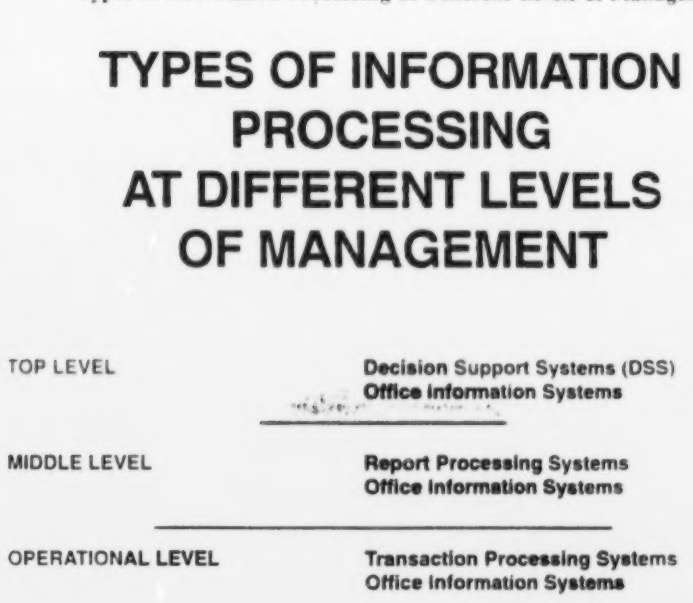
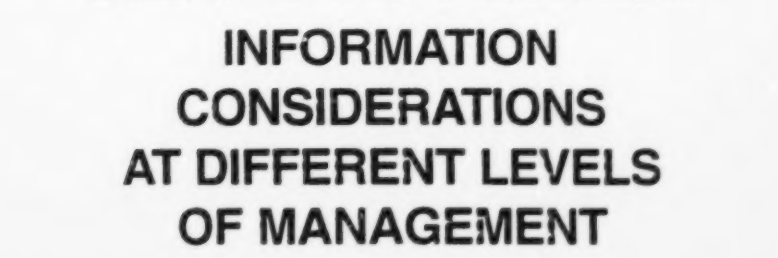


Figure 5

Information Considerations at Different Levels of Management


INFORMATION
CONSIDERATIONS
AT DIFFERENT LEVELS
OF MANAGEMENT
TOP LEVEL
Highly Summarized. Source External to firm, Ad Hoc. By Exception. Unscheduled Low Volume
MIDDLE LEVEL
Reports, Source Internal to firm. Somewhat Summarized. Periodic (Daily. Weekly. etc.)
OPERATIONAL LEVEL
Highly Detailed. Periodic. Online. Immediate Access. High Volume

Printed Materials

Many excellent textbooks for computer applications are available. Often textbooks will have accompanying study guides or workbooks, and many textbooks provide software in complete or educational versions. Textbooks should be chosen with an eye toward the future. With technology changing so rapidly, textbooks for computer applications may be outdated when they are published! Generally, the more generic the text is in its applications, the longer it will be suitable. When textbooks provide instructions for specific software, teachers must expect to change textbooks when they change software.

The readability level of the textbook and its durability should be considered. If students purchase textbooks, spiral-bound, consumable texts are the best choice for hands-on classes. Students can fill in the information on exercises, self-help tests, and other items provided in these texts. On the other hand, if the text will be provided on a rental basis, a hardbound text with supplemental teaching materials would be more suitable.

Teachers can evaluate teaching materials by using a weighting system such as the one shown in Figure 6. First, the important categories are determined. Various weights can be assigned to each category to differentiate importance. Next, the important criteria for each category are determined, and a value is assigned to each criterion. For example, consider the category of Content in Figure 6. A weight of 10 has been assigned. Each criterion in the category has been scored on a scale of 1 (low) to 10 (high). The evaluation in this category considers all criteria to be of equal importance. Thus, this category receives a score of 50. Finally, the score of 50 would be multiplied by the assigned weight of 10, giving a total score of 500 for Content. This process would be completed for all categories. Total scores for all categories would be added for an overall measure of the suitability of a teaching aid.

Software

Interdependence of hardware and software is a major consideration. Database and spreadsheet packages are usually marketed in versions or releases of versions, which usually upgrade features of the software. Upgrades traditionally include capabilities which almost always require greater hardware resources, such as more memory or disk capacity, so budgeting for software upgrades must be made based on hardware resources.

When making database and spreadsheet software purchases, whether separately or with a textbook, teachers may have difficulty deciding the brand to purchase. Consult sources of information on software popularity and performance such as the following: Trade publications such as Byte and PC World conduct

Figure 6

Textbook Selection Form

TEXTBOOK SELECTION FORM

Text Title:

Author(s)

ISBN:

Evaluator:

Directions:

1. **Give each of the listed items within a category a score 1 (lowest score) to 10 (highest score).**
2. **Add the scores for each category and multiply the results by the weight assigned to each category.**
3. **Total the weighted scores at bottom of the form.**

Category 1--Content (Category Weight = 10)

Readability is appropriate	10
Students have prerequisite knowledge or skills for content	10
Problems are relevant	10
Situations posed are appropriate	10
Scope of material is appropriate	10
Category Total	(50) x 10= 500

Category 2--Durability and Appearance (Category Weight = 8)

Item is suitable for constant use	_____
Accompanying supplements are good	_____
Quality of text is good	_____
Layout is good	_____
Category Total	(_____) x 8= _____

Category 3--Publication & Cost (Category Weight = 5)

Publication date is current	_____
Reputation of publisher service is good	_____
Author is a known expert in the field	_____
Cost is reasonable	_____
Category Total	(_____) x 5= _____

Overall Text Rating: _____

15 BEST COPY AVAILABLE

frequent reviews of software. Trade publications for the office industry, such as Managing Office Technology, The Office, and Office Systems, also frequently review software. User groups and professional industry and academic organizations, such as the Association of Records Managers and Administrators (ARMA), are helpful also. Often consultants or vendors are members of business education advisory boards. They can provide a wealth of information and sometimes provide software on a trial basis. When the time comes to purchase database and spreadsheet software, teachers should make use of the educational discounts that are available for all educational materials, including software.

Site License Agreements

Schools generally purchase software for use on several computers, and vendors provide site licenses which allow the school to make a specified number of copies of software and the accompanying documentation within a single installation or site. These licenses vary greatly in their scope and in the details of the agreement. There are restrictions on software use with site licenses. Generally, the software can be used on site only and may not be copied for use elsewhere. Copying software for personal use or beyond the site license provisions is illegal. If a teacher or student does so, the teacher, the student, and the school may be liable to prosecution.

Teaching ethical use of software and hardware is a responsibility of all computer teachers. Current laws on ethical use of database files and confidentiality should be stressed (for example, the right to know versus privacy).

Public Domain Software and Shareware

Public domain software is available to the public at no charge and is obtained through computer bulletin boards or is advertised in computer magazines. An author can retain copyright and still allow the software into the public domain by simply stating this in a licensing agreement. This condition makes the software available for use and free duplication, but not for sale by anyone but the author. The major problem in using public domain software for educational purposes is lack of support. The packages may have been updated and improved by numerous individuals with no one having the responsibility or obligation for updates or debugging. These files are a good source of viruses. Therefore, teach students how to detect and clean viruses.

Shareware software evolved from a marketing concept whereby software could be sampled and paid for only if the user found it useful. Shareware may contain files that can be used as examples or to teach database query. Under a shareware system, the author includes information on the disk(s) about the fee for

registration and to what the user will be entitled. If the user decides to use the software, the user submits the registration fee and typically is guaranteed update information, additional documentation, or a more substantial version of the program.

TECHNOLOGY FOR TEACHING

Inexpensive equipment is now available to connect a computer to a regular television through an adapter (scan converter). The television effectively becomes the display with some loss of resolution. Generally, viewing does not require a darkened room, so television projection can work very well for software demonstrations when monitors are used.

A multimedia projection system provides computer projection and video cassette projection. A ceiling projection system yields a movie-screen size image suitable for classrooms or auditoriums. The system can be connected to a computer or to videotape equipment to display a screen image, or it can be connected to other cable devices. Figure 7 illustrates a possible projected image that shows how to enter labels into Lotus. Software such as PowerPoint, Corel Draw, or DrawPerfect may be used to generate teacher-made projector overheads.

Courseware authoring systems are software packages that allow teachers to create tutorials and other on-screen materials. Toolbook and Authorware are examples of such software. Toolbook allows teachers to develop chronologically lists of tasks (reveals), animate a concept or figure, and incorporate photographs or video clips into presentations. The challenges are the time needed to master the software and hardware, the high cost of equipment and software, and production time. However, the classroom teaching and learning rewards are great.

TEACHING ENVIRONMENTS, STRATEGIES, AND EVALUATION

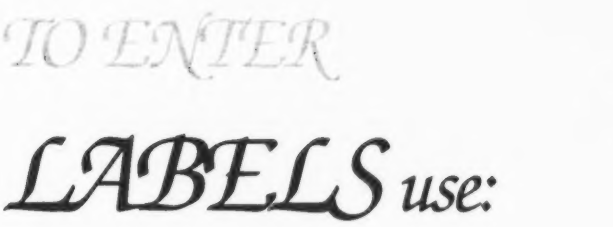
Available resources differ among schools. Also, students learn differently, so planning for student learning precedes selection of teaching strategies. Additionally, evaluation usually takes the form of "hands-on" assignments as well as written and performance tests.

Teaching Environments

Teachers may have computer labs for all class sessions, may share a lab with other teachers, or may not have a lab for any in-class work. Different learning environments pose different opportunities and challenges for both teachers and students.

Figure 7

Possible Projected Image



Any letter

A space bar (space)

A back slash \ to repeat characters

A caret ^ to CENTER

An apostrophe to LEFT Justify

A quotation to RIGHT Justify

NOTE: ALL LABELS TAKE ON A ZERO VALUE

Teaching in the Computer Lab

Computer labs allow teachers to demonstrate and students to have hands-on practice and immediate feedback. Yet, with equipment in the room teachers may have difficulty keeping everyone's attention, and students may have difficulty seeing around or over the equipment. Another problem that emerges when students are working on the computer while the teacher gives instructions is the tendency toward reaction without thought. All approaches to teaching computer software should avoid rote learning in favor of conceptually based problem solving.

To avoid these problems, the teacher can follow these procedures:

1. Have students turn off their monitors during demonstrations and focus their attention on the demonstration.
2. Teach a concept, then provide hands-on work using that concept. Require students to define the procedures they used.
3. Assign problems that require critical thinking. Have students work in groups on some problems and individually on others.
4. To prevent problems with equipment at individual student stations, encourage students to team with each other temporarily if their equipment or software fails.
5. Plan for individual learning activities. Avoid extensive large-group instruction.
6. If students are following teacher instructions step by step, be aware when several students appear lost. Repeat the procedure as needed.
7. Encourage students to experiment. If a command fails, check the procedure and try again.
8. Provide learning checkpoints where students compare their work with a model.
9. Provide for learner "fatigue." Use "time-out" activities to vary the pace and change the focus from the keyboard to problem solving. For example, play musical chairs; students move from station to station keying in one data record for a database file. At the end of the exercise, all students have a database file for further instruction.

Teaching in the Classroom without Computers

Teaching computer applications without computers seems like teaching swimming without water, but it is possible (and even desirable) to teach database and spreadsheet concepts in a traditional classroom setting. The appropriateness of this method depends to a great extent on the objectives of the course. When the primary objective is to develop a skill using software, a hands-on approach is

better. Also, when students are preparing for specific jobs, learning should take place on the computer. At the post-secondary and college/university levels, where courses stress concepts as well as skill building, instructors may use the lecture and discussion approach or a lecture/lab combination.

Two major problems in teaching database or spreadsheet software in classes without computers are the lack of student involvement and the lack of planned opportunities for students to develop proficiency. A grouping approach can help with the first situation. Students can be formed into small groups with a group leader. Within these groups, problems can be discussed and solutions presented in writing or orally by one member of the group. Students soon grow tired of viewing a screen and hearing an explanation about it without being able to actually use a computer, so the classroom without equipment should be used only to teach concepts or demonstrate. One possibility is to guide a student through the demonstration while the other students watch. Another is to have students enter data into a database file at the demonstration station. For example, six students could enter six records to form a database file. Such a demonstration emphasizes the need for data protocol. Figures 8 and 9 illustrate the need for establishing protocol or rules for data entry (using last name, and consistency in upper and lower case, for example) for spreadsheets and databases. Figure 8 could be used to review conceptual understanding of database structure. A third procedure is the use of paper-and-pencil learning activities. Figure 9 is an example of a paper and pencil review of students' understanding of how a spreadsheet treats cell entries. A combination of classroom teaching and scheduled laboratory sessions can be the best of both worlds for teaching spreadsheet and database applications.

Classroom/Laboratory Combinations

The combination approach requires planning to accommodate all students and scheduling considerations. An approach that accommodates a large number of students in one classroom may require lab assistants to manage scheduled lab periods. Another approach is to schedule students into classroom and lab on alternate days, but this method usually requires the cooperation of two or more teachers so that lab facilities are available when everyone needs them. A third approach is to schedule classes (or sections) in a regular classroom throughout the day and have teachers share the lab, working out scheduled times among themselves. In other words, if three teachers teach a class at 10 a.m., they would decide on which days they want to have their classes use the lab. They also need to establish procedures for turning in lab assignments and providing feedback to students.

In any lab configuration, the teacher must provide lab assignments that require specific outcomes. Figure 10 is an example of an assignment that requires three printouts.

Figure 8

Database Protocol EXercise

Instructions: Using the data descibed in the first column, complete the columns to the right. We will discuss your choices upon completion of this exercise.

Data Description	**Valid Field Name**	***Type of data**	**Size of Field**	****Example**
Zip Code Plus Four				
Street Address				
Name of Person				
Date of Birth				
Person has Paid Bill				
Product Code				
Vehicle ID Number				
Cost of a CD				
Hours Worked				
Price of New Car				
State Code				
Year				
Name of Product				
Quantity on Hand				
Shoe Size				
Shirt Size				

*Character, Numeric, Date, Logical, or Memo
**How would you enter the data?

Figure 9

Spreadsheet Protocol Exercise

Spreadsheet Cell Entry Exercise

Instructions: The column to the left has an entry that has been entered into a spreadsheet Cell. Use the letter in the table to the right to match how the program handles the entry.

Data Entered into a Cell		How the Spreadsheet Treats the Entry:	
____ 1.	'1995	L.	Label
____ 2.	B6+B7	F.	Formula or Function
____ 3.	@MAX(C5 to D5)	V.	Value
____ 4.	ACCOUNT NUMBER	E.	Triggers an error or keyboard lockup
____ 5.	\=		
____ 6.	654.22		
____ 7.	512-22-3388		
____ 8.	+A1(C2)		
____ 9.	+A5*B4		
____ 10.	^QUANTITY		
____ 11.	5.50 *(spaces precede 5.50 to line up values)*		
____ 12.	@NOW		
____ 13.	AT SUM		
____ 14.	+5*((C1+C2)/2		
____ 15.	1996		

28 BEST COPY AVAILABLE

Figure 10

Spreadsheet Problem with Specified Outcomes

Lotus Spreadsheet Problem:

PURPOSE: The purpose of this problem is to develop your skill and conceptual knowledge of formulas and @IF statements in Lotus. You may need to review the following before performing this problem:

Changing Column Widths
Use of the /COPY command
Procedures for using the /RANGE FORMAT commands
Use of the \- and \= for single and double lines
Sorting data using the /DATA command
Use of the @IF statement, the @NOW and the @SUM functions
Printing text of formulas

INSTRUCTIONS:

Enter the spreadsheet data given below: (NOTE: your row and column addresses may vary..these are just suggestions):

	A	B	C	D
1		Your Name Payroll		
2		*Current Date (@now)*		
3				
4			PAY	GROSS
6	NAME	HOURS	RATE	PAY
7	==========	==========	==========	==========
8	DAVIS, JENNIFER	40	5	
9	ARONSOHN, AGNES	38	4.5	
10	ZIGLER, KAREN	40	12.5	
11	ADAMS, WILLIAM	25	8.55	
12	GIBBS, ANDREW	40	6.25	
13	MURPHY, AUDI	40	7	
14	SMITH, TERRY	25	12.5	
15	BENSON, SUSAN	40	10	
16	ZIMBLOSKI, TONY	40	11.5	
17	----------	----------	----------	----------
18	TOTALS			
19	==========	==========	==========	==========

1. After entering the above, format data columns and save the file as **PAYROLL1** on your diskette.
2. Using **only the hours worked and pay rate,** compute the gross pay column using an @IF statement. Overtime pay is computed on a time and a half basis. Be sure to check your calculations to ensure their accuracy.
3. Provide totals for the hours worked and gross pay columns.
4. Save and replace the file and **PRINT**.
5. **PRINT** the file using the **TEXT OF FORMULAS**.
6. Save and replace the file and sort the spreadsheet by the name column. Check the results and **PRINT** the sorted spreadsheet. Save the file after sorting as **PAYROLLS**.

The rules of good pedagogy dictate sensitivity to students' learning styles and application of the principles of learning. Some students work well alone; others prefer to work in teams. Some students prefer reading instructions; others learn from software demonstrations or screen illustrations. In addition, teachers must consider students with special needs. Several types of assessment instruments are available that teachers can use to determine the learning style of a student. Most require special training for their use or assistance from counselors or colleagues who are trained in their use and interpretation. Using these instruments can be helpful in planning student activities for spreadsheet and database classes, especially when computers are not available to all students at the same time.

Students should see the relevance of what they are learning. They also should be able to apply what was learned to new, similar situations; they should be actively involved in the learning process.

Problem Solving

Teaching students to solve problems with software tools is the ultimate goal of any teaching methodology employed in spreadsheet and database applications courses (Nellermoe, 1993). For example, students should understand the distinction between designing their own database and using a database management system that has already been designed for a specific purpose, such as keeping track of printed records.

The problem-solving approach to teaching spreadsheet and database applications requires students to become familiar with the software (i.e., gain knowledge) and to understand the features well enough to use them in a variety of situations. Ultimately, the student applies the learning to an unfamiliar situation by analyzing the elements of a business problem, deciding what is relevant to solving it, selecting the appropriate features of the software, and solving the problem. If the spreadsheet or database software package being used is not suitable for solving a particular problem, then students should have enough knowledge and skill to choose one that is better. They should be able to adapt or revise files to fit new situations and to generate new information that was not previously required.

Obviously, a student must learn the necessary software features to become a capable problem solver, but masterful operation of the software does not guarantee the ability to understand and solve problems. A step-by-step approach to problem solving can be used with database or spreadsheet software. Figure 11 illustrates spreadsheet problems that apply the problem-solving approach.

1. Define the problem. What problem needs to be solved? For example, compute overtime pay in a spreadsheet (Figure 11).

2. Identify the factors of the problem. What must be known about pay rate and hours worked?

3. Determine the means of transforming and interpreting data. What features should be used to solve the problem? In this example, the student must know how to use formulas and the @IF statement correctly.

4. Develop criteria for selection. What criteria must be met to solve the problem? In Figure 11, the student must provide a way to check whether a worker has overtime hours.

5. Gather the interpret data. What resources can be used for gathering information? How can we extract the necessary information? On-line help files, a manual, texbook, and class notes are resources.

6. Draw conclusion(s) and make recommendations. The student must check the reasonableness of the spreadsheet to be sure the data are logical. Too often both teachers and students forget that the objective is to use the software productively in a realistic setting, not just to become proficient in executing the functions.

A subset of problem solving is critical thinking, which involves distinguishing between facts and assumptions, determining the reliability of sources, verifying the accuracy of data, distinguishing between relevant and irrelevant information, detecting bias, recognizing logical inconsistencies, and determining the strength of an argument (Maxam, 1990). Teachers should help students understand and use these thinking skills before they tackle a problem using database or spreadsheet software.

Graphics should be an integral part of spreadsheet knowledge. Part of the problem-solving process in spreadsheets should include preparing charts and graphs that are appropriate illustrations for the data entered into the spreadsheet. Students should be taught how to prepare charts and graphs with the particular spreadsheet package being used. Moving from simple to complex, adding more sophisticated features to the graphs and charts, and applying their knowledge to developing the best illustration of the data represents the students' problem-solving abilities.

The problem-solving process can lead students to visualize ways in which files from one software package can be exported to other packages. They can then see the value of software packages beyond preparing spreadsheet and database files. How one package relates to another, how the business world transfers information among packages, and how data thus become information for decision making are important applications of the decision-making process.

Demonstration

The underlying principle of the demonstration method is that one must learn to use a tool before the tool can be used effectively. When teachers use this

approach, they teach students the basics of the software first in a lab setting and then have them apply the features of spreadsheet and database software to solving problems. The focus tends to be on learning the software rather than solving problems. Classes are taught in a computer lab with the teacher showing various software functions and ensuring that students execute the functions correctly. Students work simultaneously at their workstations while the teacher demonstrates. This method works well under most circumstances but requires some checking to be sure that the students are achieving the results being demonstrated.

Time is a deterrent in demonstrations. An inherent danger is that the material covered will be determined by the slowest students. If it becomes difficult to keep the entire class on track, it may be necessary to tell those who fall far behind to watch their partner's screen rather than attempt to catch up. Alternatively, an able student can be trained as a lab assistant to move around the classroom and help students who are having problems.

In a demonstration, it is important to encourage students to help each other. Allowing students to talk and help each other can be very effective and reduces the need for the teacher to deal with every glitch that occurs. Normally in a class of beginners, several students will encounter some difficulty during the demonstration; therefore, having students develop a buddy system can be beneficial.

Another form of demonstration is more teacher centered and may take place in a lab, but is most often applied in a classroom without computers. This form of teacher demonstration provides basic information, but students do not perform the functions as they watch. Overhead transparencies and/or computer projection equipment are used to demonstrate the software, and students complete assignments individually in a computer lab. Students may practice what they have seen demonstrated by completing a tutorial or a series of assignments. Since the students are not using each feature as it is demonstrated, the teacher does not have equipment and software problems to contend with while demonstrating. Students should always have access to a computer to apply the features of database or spreadsheet software, but they do not always need that access during a class period.

Figures 11 and 12 involve students in the learning process and can be used in a computer lab with students working along with the teacher or in a classroom without computers.

Simulations

Simulations offer many opportunities for students to have fun, be involved, and solve problems. A simulation is a replication of a real-life situation. Simulations can take the forms of role playing and models. Role-playing simulations require the participants to occupy positions in a hypothetical setting,

Figure 11

Problem-Solving Approach in Spreadsheet Problem

THE *OT- CALCULATION* QUESTION?

SPREADSHEET:

THE PROBLEM:

Develop a spreadsheet layout that will allow a user to compute overtime pay for a company. Determine what variables will be needed to compute overtime and how you would set up the spreadsheet. The "normal" calculation for overtime is time and a half for any hours over 40. Consider separating hours into regular and overtime hours, or using a formula that just considers the total hours worked. Using only total hours would require the use of an @IF statement. Consider situations where a person may get overtime for work of less than 40 hours during a week.

Figure 12

Problem-Solving Approach in Database Problem

THE *MUNCH-O-MATIC* EATERY

DATABASE:

THE PROBLEM:

Develop a database that will allow a user to access information about restaurants in your community. Information may include name, location, type of menu, type of facilities (eat in, carry-out, etc.), to name a few. You decide what you want to include. Once you have decided what fields you will include in your database, create a database file structure and enter information for at least 10 eateries in your community. Create documentation that allows the user of the system to add, delete and change records. Also consider adding a section that will allow the user to add and delete information fields.

such as a law office, where they perform the tasks required for the position. This type of simulation allows students to interact in situations similar to those they will encounter in real life and to perform the tasks that employers expect. The model simulation incorporates role playing in a sophisticated model that represents the real situation as closely as possible. Generally, office simulations employ many different types of software, including database and spreadsheet packages. Office simulations are available from a number of educational publishers.

Self-Paced Learning Modules

A module is a teacher-designed unit that meets whatever objectives the teacher wishes. A module usually includes performance objectives, self-tests, learning assignments, and an assessment instrument to measure overall achievement. Since the tutorials provided with most database and spreadsheet software have structured lessons, these could be used for the assignments and self-test portions of the module. In addition, since students work at their own pace, some system is necessary to keep track of their progress; and someone must develop procedures for evaluation. The module approach can be a choice for remedial work, for students who cannot take a scheduled class for database of spreadsheet applications, or for open-entry/open-exit programs.

EVALUATION OF PERFORMANCE

Testing can be divided into two general categories, performance and objective, but evaluation takes many forms. Evaluation implies judgment as well as measurement. Several evaluation approaches are appropriate for classes in which software is used and students are expected to learn database or spreadsheet features. Pretesting and observing student behavior can be used to determine readiness; homework and in-class exercises can provide interim measures of student learning; recitation (and demonstration) are useful to measure students' ability to convey ideas; individual and group projects as well as simulations can be used to measure initiative and teamwork as well as affective behavior; performance tests measure the students' ability to use their skills and to produce (sometimes under timed conditions); and objective tests measure achievement primarily in the cognitive domain (Schrag & Poland, 1987).

Pretesting and Observation

Pretesting and/or observing student behavior is an important first step in evaluating learning, especially in classes that require prerequisite skill or knowledge. Sometimes a course prerequisite is used to ensure student readiness, but that may not prevent unprepared students from entering the course. Often a

class will exhibit substantial variance in student knowledge and skills, a situation that will frustrate student learning unless the class is pretested.

Pretests can be simple timed writings if keyboard skill is needed or a more elaborate measure which may be a form of the post-test that will be given at the end of the course. In a spreadsheet or database software course, a pretest on basic math or computer literacy would be useful in a beginning course. For advanced courses, a pretest would measure knowledge (or skills) that the student should have acquired in earlier courses using the spreadsheet or database software. Pretesting accomplishes two objectives: 1) students who are not prepared for the course can be given remedial work or routed to lower-level courses, and 2) students who score sufficiently high on the pretest can be enrolled in a higher-level course. In a skill-oriented class, when pretesting is not possible or practical, simply observing students at the computer may provide sufficient information to judge student readiness for the course.

Homework and In-Class Assignments

Homework in computer classes often takes the form of completing a computer assignment and turning in the hard copy results. Teachers should find ample assignments in textbooks to have students complete in class or in a lab. In-class assignments can be valuable to guarantee that certain skills are accomplished, but there is one serious drawback associated with in-class computer assignments. There may be students who will complete the assignment much quicker than the rest of the class, and there may be students who will require the majority of the teacher's time to complete the assignment within the class period. When in-class assignments are used, some provision must be made for these students. Extra, more challenging assignments can be provided for those students who finish quickly or those students can tutor other students who are having a difficult time. Without careful planning, in-class assignments can work against the self-paced approach that is often productive in computer classes.

Here are some procedures for managing in-class work and homework:

1. Include unique information in computer assignments, such as the student's name, address, social security number, date and other instructor-assigned data.
2. Select assignments that require a sequence of editing or printing so that the last output has evolved from the first output. Of course, do not suggest that the student save all of the intermediate steps.
3. Require "Print Screen" output of the student's disk directory to show the names of the files with their date and time information as part of assignments.
4. Have students include their identification as part of the data (for example, Smith Company as a heading in a spreadsheet for a student named Smith).

5. Items that are part of the homework assignment should relate to tests, so that if a student does the homework, he/she should do well on the test.

Grading of computer assignments tends to be more straight-forward than grading in other types of courses. Complex assignments can be modularized so that points are taken off for each element that is not completed correctly. Another approach is to set only one or two objectives for a complex assignment and ignore other elements. Teachers should consider whether homework and in-class assignments are a major or minor part of the total evaluation. If they are a small component, not all assignments would necessarily be graded.

Objective and Performance Tests

Tests can be classified as norm-referenced tests and criterion-referenced tests. Criterion-referenced tests are tests that are used to determine an individual's status as it applies to a well-defined behavior domain. Norm-referenced tests are constructed to produce information about an individual's performance relative to others taking the exam (Stehr, 1981).

Objective tests are composed of items that are scored as correct or incorrect and take the form of multiple choice, true and false, short answer, matching, or completion. A performance test requires evaluation of a product that the student has created or a process that the student performs such as creating a file structure for a database application or preparing a spreadsheet. When students are expected to exhibit a broad base of conceptual knowledge, objective tests are appropriate forms of evaluation. When successful performance of a particular task or completion of a specific product is needed to evaluate student learning, a performance (or production) test is appropriate. Both objective and performance tests have a place in database and spreadsheet evaluation. Items on an objective test, such as the ones shown in Figure 13, can measure learning effectively. However, when students are expected to be proficient both in using the features of the software and solving a database or spreadsheet problem, a performance test will be a more effective measure. Textbooks generally provide test banks for compiling objective tests. Publishers usually provide test banks on disk and test-generation software for producing customized tests. Some or all of the test bank may be used to create study guides or self-test items for student use. Another form of computer-generated testing, interactive testing, allows students to take an objective test on the computer. The program scores the responses and reports results, giving students and teacher immediate feedback.

Performance tests measure student mastery of specific skills and the ability to apply those skills in a variety of situations. When evaluation requires evidence that a student can perform, a performance test is preferable. Because performance tests require the teacher to use specific criteria for evaluation of the product, the

Figure 13

Objective Test Items

Use the following to answer the next set of test items.

PAY INFORMATION

NAME	NUM	HOURS	TOTAL
ADAMS, JOHN	101	30	$340.22
BENSON, BLAKE	102	10	$200.00
DENTON, SUE	103	8	$100.00

Task to Perform **Spreadsheet Function**

1. add another name
2. remove all totals
3. add sub-headings
4. remove a name
5. add column totals
6. move spreadsheet to right
7. move spreadsheet down page

A. close (delete) column

B. open (insert) row

C. close (delete) row

D. open (insert) column

Multiple Choice:

8. Which of the following formulas will subtract the contents of B5 from B4, divide the difference by 7, and add the result to the contents of B3?
   A) +B3+(B4-B5)/7
   B) +B3+(B4-B5)^7
   C) B3+B4-B5^7
   D) B4-B5/7+B3

9. In the formula 14+(7-(3+2)+8), what will be the first operation to be performed?
   A) 14+7
   B) 7-3
   C) 7-5
   D) 3+2

10. In the formula 4+6-2+12-8, which is performed first?
    A) 4+6
    B) 6-2
    C) 2+12
    D) 12-8

11. Cell C8 contains the formula +B4-B6. B4 currently contains the value 10, and B6 the value 3. What will be displayed in C8 when the contents of B6 are changed to 8?
    A) 7
    B) +10-8
    C) 2
    D) 5

criteria should be clear. The criteria should be explained to students so they will understand what is expected, and students should be given a trial run before the actual test to familiarize them with the test procedures and allow them to practice working under time pressure.

Evaluating the student files stored on disks, the teacher can evaluate the processes used to solve the problem as well as the product. For example, on a spreadsheet test, the teacher can examine a file to determine whether proper formulas were used for calculations. On a database test, the teacher can check to see that students created the proper types of files to solve the problem.

Testing on the computer can cause an inaccurate measure of student performance if the equipment malfunctions. Therefore, the teacher must anticipate possible problems with equipment and plan for alternatives.

Students should be acclimated to computerized performance evaluation early in the course so they are less anxious about the process at the time of testing. For example, practice tests, which do not count heavily in their grades, can be given as frequently as needed. A number of tests should be scheduled throughout the course so that a single test is not a major factor in success or failure.

To determine the unit grade or the course grade, the evaluation could include in-class/homework assignments, practice problems in preparation for the test, a performance test, and an objective test. For example, a teacher might assign 6 in-class projects worth 1 point each and 8 assignments worth 3 points each. A student who completed all 14 assignments with no errors could earn a maximum of 30 points. Students who do not successfully complete the assignments should be given individual help so they are prepared for the practice test.

The practice performance test would be given to test student understanding of major concepts and to prepare them for the summative evaluation. For example, students may be assigned three problems worth 15 points over a spreadsheet unit. A student who completes the problems with no errors could earn 45 points. Students who do not do well on the practice test can be given additional help before the final performance and objective tests on that unit.

The final performance test could be weighted at 75 points. That test should cover the major concepts taught in the unit and practiced by students in homework and in-class assignments and the practice test. If students are furnished with a test diskette, the files stored on disk can be checked, as well as printouts. Part of the evaluation can be based on proper applications evident on the disk file.

A final component of grades would be a theory test that could be assigned 50 points. The final grade would be based on these factors:

30 In-class/homework assignments
45 Practice test
75 Performance test
<u>50</u> Theory test
200 TOTAL POINTS POSSIBLE

SUMMARY

In spreadsheet and database units, students must learn to think critically, to plan spreadsheet and database files, and to determine how the results are to be used. These processes are much more important than just "pushing the keys" to enter data. Therefore, the teacher has a major responsibility of assisting students in developing their problem-solving skills.

The teacher also must choose carefully among the myriad of teaching materials and tools available, must carefully plan content and activities to achieve desired results, and must assess performance to be sure those results are achieved. Teachers must realistically plan for individual learning styles, readiness for the course content, and individual differences among students and their ability to learn. These variables make the task of a computer applications teacher both a challenge and an opportunity.

SPREADSHEET AND DATABASE SOFTWARE REFERENCES

Cringley, R. X. (1992) Accidental Empires. New York: Addison-Wesley Publishing Company.

Illinois State Board of Education. (1991). Tech prep planning strategies. Springfield, IL: Dept. of Adult, Vocational and Technical Education.

Lomo-David, E. (June 1989). A comparison of spreadsheets usage in management information systems (MIS) departments of selected U.S. top 1,000 companies and spreadsheet courses taught in MIS departments of member institutions of the American Assembly of Collegiate Schools of Business. Unpublished doctoral dissertation, Memphis State University.

Maxam, S. (February 1990). Teaching higher-order thinking skills: The need for research, Business Education Forum, 31-35.

Nellermoe, D. A. (May/June 1993). Problem solving in software tool courses: Beyond the basics, Journal of Education for Business, 65(5), 273-275.

Rathswohl, E. J. (Summer 1990). Information resource management and the end user: Some implications for education. Information Resources Management Journal, 3(3), 2-7.

Schrag, A. F. and Poland, R. P. (1987). A system for teaching business education, 2nd ed., New York: McGraw-Hill Book Co.

Stehr, B. W. (1981). Effective evaluation in business education. St. Peter, MN: Delta Pi Epsilon.

TEACHING WORD PROCESSING AND DESKTOP PUBLISHING

by
George A. Mundrake

INTRODUCTION

When asked to agree on the most widely utilized computer software, most people would answer "word processing." Not so long ago this would not have been the case. During the early 1980's most computer users were struggling with complex programming in BASIC to perform tasks on their computers. To make applications software easier for the user, hardware must be more sophisticated. If the hardware is more advanced, easier software can and will be produced.

If one were to make an analogy of advances in computer technology to an earthquake, word processing technology for computers would be the tidal wave caused by these advances. Word processing software advancements illustrate the trend to make computing easier for the user through software. As the hardware has become more powerful, the more word processing functions can be performed in a more user-friendly manner.

The purpose of this publication is two-fold. The first goal is to assess where we are in the areas of word processing and desktop publishing, and the second is to suggest strategies and methods for instruction. As with any changing technology, periodic and systematic evaluation is needed to provide better instructional strategies for our students.

TECHNOLOGICAL INFLUENCES AND THE CURRICULUM

For years technology has touched our curriculum and the ways we teach. Concerns over what and how to teach are changed by the available teaching resources. Research has brought us programmed learning, teaching machines, teaching through radio and television, and other cycles of technology that have influenced how and what we teach.

How we accept and use the technology in education depends upon how our society accepts and uses the technology. Society has widely accepted the use of word processing and desktop publishing. At the same time society wants students to be able to perform the basics and have the ability to adapt to new environments. These provide unique challenges for educators. To understand the influence of technology requires a more careful look at how technology has evolved in the areas of word processing and desktop publishing.

Technological Advancements

In the early years of microcomputers, emphasis was on teaching BASIC programming and using few canned applications software or spreadsheets. Few applications packages were available for microcomputers. The word processing task was relegated to dedicated word processors like WANG and CPT, and IBM minicomputers with DisplayWrite. Today the word processing function is clouded by endless versions of very sophisticated software operating in multiple computer environments. Since the first microcomputers, we have been through at least five generations of hardware configurations in both the DOS-based and Apple computer operating systems. As an example, WordPerfect software has evolved through versions for Macintosh, DOS, Microsoft Windows and for mainframe computers such as Digital's VAX.

The evolution of word processors may also be characterized by the way the user interacts with the package through the hardware. Earliest word processors were key driven (without the function keys). The user must memorize many keystrokes or refer to a desk full of templates to perform tasks such as "Control/U" to begin and end underlining. Next, more user-friendly versions used a "hot-key" such as the escape key to bring a menu to the monitor. These were easier to learn but not designed for productivity because of the time needed for the menu to appear. Function-key-driven software then appeared, where function keys and combination of function keys performed the word processing tasks. Again, the user needed practice to remember the combinations of function keys to perform the task.

The mouse contributed to the latest software developments aimed at graphic user interface (GUI) or "gooie." The user could click and drag to block text and point at icons to perform other word processing tasks. Productivity was decreased

because the user must return the hands to the keyboard to enter text. Most keyboards now have small bumps on the "F" and "J" keys to help users find the home keys.

Using mouse and icon-driven software posed another problem for people teaching and learning word processing. The "one click does it all" ease of operation made it more important for students to know the basics of word processing. The use of one icon may invoke a series of what used to be keystrokes that perform a wide range of unexpected tasks. The user needs to know what to expect when the icon is clicked. How to reverse the process if the wrong icon has been clicked is also important.

Software vendors have assisted users by compromising and duplicating methods to perform tasks that fit the users' abilities and needs. Functions that users perform infrequently are placed in menus. Selecting printers, setting up tables, and changing the colors of the monitor are examples of these tasks. Frequently used tasks are performed most efficiently by using memorized keystrokes such as bolding, underlining, and centering.

More recent word processing programs are including desktop publishing features and allowing users to view files as WYSIWYG (What You See Is What You Get) format. These include adding rules, importing graphics, shadowing text in boxes, using dropped caps, and utilizing files (piping) from other applications through import filter software.

Probably the most significant development in hardware that makes advanced word processing and desktop publishing (DTP) possible has been the laser printer. With the laser, fonts, type styles, and design considerations become a more important part of the document quality. Without the development of the laser, desktop publishing would not have developed. Specifically designed desktop publishing software packages such as Ventura, Quark, and PageMaker performed poorly as word processors. Now word processors are becoming better desktop publishing software, and DTP software packages are becoming better word processors.

Word processing software that performs desktop publishing tasks and DTP software packages require high-end hardware. Hard disk drives with 1 gigabyte of storage are not uncommon. Random Access Memory (RAM) of eight or more megabytes is also more common. Including one graphic in a document may take up an entire floppy disk. At a resolution of 300 X 300, a four-inch-square graphic will consume over one megabyte of memory or one high-density 3 1/2-inch disk. Like word processing software, DTP software has developed quickly over the past few years. Several versions of packages are available, and there is no end in sight for the development of both hardware and software.

Multimedia hardware and software have emerged, combining word processing text, digitized or scanned graphics, sound from sound digitizing cards and microphones, and video input devices into presentations. The next step in

technology will include video, voice, and text in user-friendly formats with CD/ ROMS and erasable CD's or MO (Magneto Optical) drives at the center of hardware technology.

Have technological advances really caused an uncontrolled impact on curriculum and methods of teaching word processing and DTP? Uncertainty and a rapid rate of change have given us a sense of urgency to change everything. The same technology that allows us to use computers as tools to ease office burdens has also lured us into a belief we must continually change every aspect of our teaching. However, means of assessment, objectives, methods, and evaluation have remained the same.

What really has changed has been the content and our audience. The computer as a tool concept has shifted our emphasis from the computer field as a career to the computer as a productivity tool for everyone. The shift of computer curriculum as general education is slowly being realized at all levels. Because a primary use of computers is for word processing tasks, word processing will continue to serve as the entry level for students.

Desktop publishing has become a productivity tool for a wide variety of users. Most people who use DTP software are doing so to improve the appearance of their documents. However, as a career area, desktop publishing is also a viable vocational area. The implication is that students should have more diverse backgrounds, and methods and activities need to meet those backgrounds. Teachers need to provide a variety of experiences for the students.

A Word Processing and DTP Curriculum

Word processing has a role in the three major levels of curriculum: secondary, vocational/technical, and college/university levels. The level of coverage of any package depends on the purpose of the course. If the course is an introductory course for computer applications, coverage of word processing will probably be shared with spreadsheet and database applications. In this case, only the basics of word processing would be covered. If the course is vocational or technical, the course would go beyond the introduction of word processing to a greater level of proficiency and skill in completing tasks in a timely manner. A second or third course may be offered to hone skills, learn advanced features, and introduce students to other word processing packages so they become more marketable and versatile.

At the college/university level, word processing may also be part of an introductory course in computer concepts and/or microcomputer applications with spreadsheet and database applications. Word processing also is part of several courses for information systems students and office support personnel areas. At the college/university level, word processing as a tool for anyone is emphasized. A few years ago, people based their careers on learning dedicated

word processing systems. People in other areas of business were not inclined to learn these systems nor to enter their own text. With the coming of age of PCs, it is not uncommon for individuals to take care of their own data entry (if they can keyboard) and leave the editing to the support personnel. The closer the source, the less likely errors will occur. This shift has also helped enrollment in word processing classes of all kinds of students with all kinds of majors.

Desktop publishing as a course at the secondary school level currently may be a luxury because of the high cost of equipment and software. Another major problem is the incorporation of technology with printing and publishing conceptual content areas such as design, typography, and printing technologies. However, at vocational levels, desktop publishing has become more popular where funding is available as part of the office technology curriculum or as part of curriculum integration and interdisciplinary courses.

At the college/university level, desktop publishing as a part of the business curriculum has evolved primarily because of the availability of equipment and software. Business applications of desktop publishing for business cards, flyers, tickets, certificates, business forms, and newsletters made courses feasible. Courses or instructional units may also be available in journalism or advertising/ marketing departments as a tool for students in those majors. As in the secondary school curriculum, desktop publishing must also emphasize the whole publication process such as design, typography, and printing technology to be successful. Another success factor is flexibility in the kinds of projects designed to draw enrollment from other majors such as accounting (to generate better reports) marketing/advertising (to produce better campaigns), information systems (to create better proposals), and communications and general curriculum (to communicate through text and graphics).

TEACHING STRATEGIES AND DELIVERY

Development of teaching strategies requires a dynamic approach to curriculum development. By the nature of word processing and desktop publishing, change can be addressed by following a simple instructional cycle model. The basic parts of the instructional cycle include assessment, development of objectives, providing instructional strategies, and evaluation of outcomes.

Assessing the Learner

Assessing the learner involves the evaluation assessment of entry-level performance and learning styles preferred by students. Often classes at the introductory level have a wide variety of both performance levels and learning styles. As the courses progress, the divergence narrows for both elements. A pretest or diagnostic test can be used to assess the performance level of students;

however, the decision to use pretests or diagnostic tests is difficult. Yelon (1985) suggested a series of questions to help decide about pretesting:

1. Is evidence available on entry behavior? (If no, pretest).
2. Are resources available to pretest? (If yes, pretest).
3. Is time available to remedy deficiencies? (If no, don't pretest).
4. Are entry behaviors critical to enabling objectives? (If yes, pretest).
5. Are entry behaviors critical to other units? (If yes, pretest).
6. Is an objective, even though part of a previous course, important to the outcome of your course? (If yes, pretest).
7. Is a pretest needed for student satisfaction and expectation? (If yes, pretest).

Using these questions, the major question is the third one. Many teachers do not have the luxury of time for pretesting in beginning classes or computer applications classes where there are several packages to cover. Item number five is an important factor for pretests for skill courses. Since Tech Prep and vocational-oriented word processing courses act as building blocks for other courses, pretesting should be seriously considered.

If a pretest is used, the results may help the teacher choose instructional strategies based on the distribution of scores. Yelon (1985) suggested that with a wide range of scores and many resources available, individualized instruction should be considered. In classes with large numbers, high scores, and a few key behaviors lacking, use of whole class reviews and teacher reviews should be used. In classes with evenly split scores between those who know and don't know, peer tutors should be used.

Most of what we know about how people learn is still a mystery. Dunn (1981) found that people learn differently and are influenced by their environment (presence or absence of factors such as heat, light, etc.), their emotionality (motivation, persistence, responsibility), their sociological preferences (learn alone or in groups), and their physical preferences (energy levels, time of day they learn best). A major problem of assessment is to determine an individual's learning style preferences.

Dunn, Baudry, and Klavas (1989) summarized much of the research on learning styles and found that every student has a learning style and can learn the same material. This alone suggests that teachers provide a wide variety of instructional environments for students. Reiff (1992) reported that understanding learning styles leads to reduction of student and teacher frustration, higher student achievement and improved self-concept, accommodation of a variety of students in the classroom, versatility that is crucial to learning, and improved communication with administrators, parents, counselors, and other staff. Considering the variety of student abilities, emphasis on learning styles, and the

divergent backgrounds of students, teachers need to provide a variety of learning activities.

Curriculum development is an ongoing process that requires assessment and input from many sources. Other forms of assessment include follow-up studies of graduates to determine what they are using in the work place, surveys of businesses, and communication with teachers of advanced courses to determine future needs of students.

Establishing Objectives and Content

General course objectives should be developed at the curriculum level. This step is often confused with the content of the course. The processes within a package will change with time; however, the basic concepts and general objectives usually will not change. General objectives that are typical of word processing courses or units within a course include the ability to:

- define terms related to word processing
- identify and describe word processing features (blocking, search and
- replace, boilerplating and merging, etc.)
- perform word processing functions (hands-on)
- describe hardware and software needed for the task

Similar objectives for desktop publishing courses and units are needed. Desktop publishing also requires a background of knowledge in the publishing area. Objectives for desktop publishing include the ability to:

- define terms relating to desktop publishing
- describe the hardware and software needed for desktop publishing
- develop skills in design and layout of desktop publishing applications
- perform desktop publishing tasks using ____ software

More recent concern for lifelong learning has emphasized an awareness of tools and techniques to help users of word processors remain current in the field (especially students on a career path). An additional objective involves flexibility of the user and ability to adapt to different situations by using a critical thinking approach.

For each course or unit, a content grid should be developed. The grid is useful in integration of skills among different courses in the curriculum. Manpower Temporary Services has used such a grid in training of word processors for several years. The grid can be divided into basic skills, intermediate skills, and advanced skills. The basic skills list should include functions needed to get started with word processing, and the list could be included in a course where word processing comprises about one-third of the computer course content. The intermediate and

advanced tasks are more suitable for a complete course in word processing or several courses used for career or tech-prep curricula.

DELIVERY TECHNIQUES

Utilizing a variety of delivery techniques is probably the most important concern for teachers. A wide variety of activities provides different environments for different learning styles. Selection of delivery techniques depends on the instructor's comfort with the technique, the audience (number, learning styles, and backgrounds), the nature of the material (skills, concepts, and applications), and the resources available to the instructor (including equipment, personnel, and time available).

Research on various types of delivery systems consistently indicates no significant difference in student performance among delivery systems; however, most research does not give students a choice of what group they will be in, nor has there been much attention to the quantitative effect of a variety of activities on learning because of the difficulty in isolating treatments and previously learned behaviors. Orr (1990) studied two methods of teaching word processing (DisplayWrite) including a traditional method (following step-by-step printed material) and a computer-based tutorial. No significant difference was reported between these methods on student performance. Likewise, Pollard and Kizzier (1992) found similar results when comparing interactive videodisc instruction and videotape instruction. In their study, both were reported as viable vehicles for learning WordPerfect 5.0.

Lecture and Demonstration

Lecture techniques have long been criticized as being a "last resort" for teaching. The arguments against this technique are further enhanced by the nature of the audience and the subject matter. Working with word processing and desktop publishing is a "hands-on" subject matter; however, there are some content areas of word processing and desktop publishing that may well be suited to the lecture technique. These include content areas that deal with hardware, software terminology and concepts, letter/report format guidelines, and desktop publishing design concepts.

With large audiences and limited resources as factors to consider, lecturing or variations of lecture techniques can be used for reviewing new material, pre-session preparation (advance organizers), vocabulary, and description of concepts. In desktop publishing classes, lectures with good visual aids provide a good forum for teaching design and typography.

Possible major problems with lectures include the quality of the presentation and the lack of student activity in the learning process. Quality can be improved

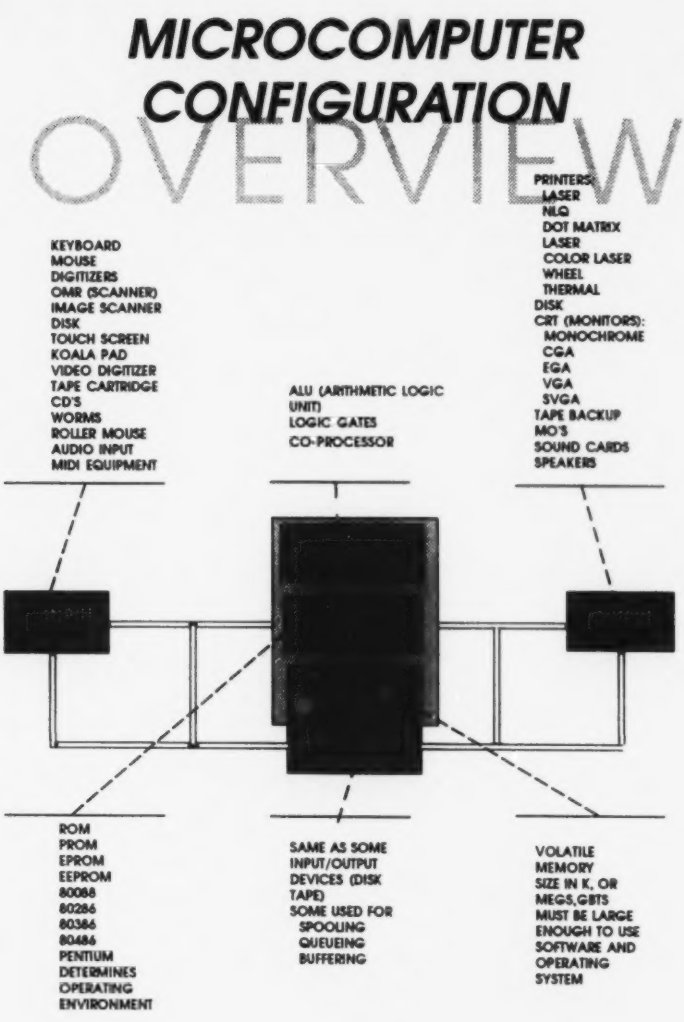
through better planning and preparation, better use of visual aids, and more interaction with teachers and students. The lack of activity can be reduced by involving the learner through note-taking, advance organizers, and fill-in handouts. One form of the advance organizer is the "expository organizer." According to Newell (1984), the expository organizer is designed to provide general superordinate concepts and principles which are already familiar to the learner and which will provide a bridge to the new material to be learned. Figure 1 illustrates a simple form of an advance organizer where students can add their own notes. Another example of this type of organizer is to ask students how music is recorded on a magnetic tape. Students will understand that the tape is sequential, and it is impossible to insert something on the tape without re-recording it to the tape. This concept could be used to show how hidden codes work in word processing programs. The same organizer can also help to explain the blocking feature of most word processors.

Demonstrations are excellent methods for teaching computer classes if the demonstrations can be seen by the students. Good quality projection systems make the demonstrations visible for students. The key to a successful demonstration is to try the demonstration in advance to make sure it will work! If the demonstration fails, a quick recovery is important. The "recovery" may be in the form of another "mistake." Failure in a demonstration, such as leaving a disk out of drive during saving, may act as a learning tool. It gives the instructor the opportunity to ask students "What happened?" or "What do I do now?" Failure in the demonstration may not be as harmful to learning as the failure of the teacher to recover. A simple "no problem" attitude may impart a "can do" attitude to the students; however, don't overwork the "strategy." Other points to consider about demonstrations include:

* Consider length of demonstrations--more than 20 minutes is long.
* Include advance organizers--tell the students up front what they should be learning and why.
* Sequence the material in logical order--easy to difficult, and general to specific tasks (such as line and page formatting).
* Where possible, show keys, menus, or indicate what your hands are doing. The demonstration is not a magic show.
* Illustrate alternative methods to gain the same output and give reasons why, or demonstrate preferred methods for doing a particular task the way you did.
* Illustrate "what if" scenarios when possible (as a closing activity) to show how to recover from errors along the way. For example, if a student goes into line format rather than page format, show how to switch to the correct format choice.

Figure 1

Advance Organizer



BEST COPY AVAILABLE

Self-Paced Materials

Self-paced materials may take many media formats: linear or branched printed materials, sound tapes, video tapes, videodiscs, computer-based instruction, on-line word processing and desktop publishing program tutorials (with a program), and printed learning activities that are performed on a self-paced basis. A number of "canned" packages are available for word processing and desktop publishing instruction, including tutorials provided by the vendor. Major considerations for their use are the resources available for their use, the numbers of students involved, the material they cover, and the homogeneity of the audience in terms of abilities. If resources are abundant, abilities are diverse, and numbers are small, self-paced materials are indicated. Since research shows no significant differences among these strategies, their use is appropriate.

Teachers should be concerned with record keeping and scheduling of self-paced material. Rotation schedules, completion check lists, and performance evaluations along the way are important to retain control of the learning environment and to provide consistency for the learner. Can self-paced materials be used with larger groups? Yes, because work we do in word processing and desktop publishing classes is by nature self-paced. Individual problems are completed at different times by students. Also, the numbers of problems completed are different because students are different. The design of self-paced learning activities is critical for the teacher as well as the learner. For the teacher, the design should fit in with the content, should provide measurable outcomes, and should allow the teacher to maintain control. Students need consistency, purpose, and sets of procedures to perform the tasks.

Problem-Solving Learning Activities

Word processing and desktop publishing are areas where problem solving and critical thinking are important learning techniques. Problem solving includes determining a problem and generating solutions. Problem-solving skills have become a trait required by employers over many other technical skills as reported by Everett and O'Neil (1990). Nellermoe (1993) and Hunter (1992) also stressed the importance of students' ability to adapt to different situations through problem solving and critical thinking. Critical thinking includes analysis and evaluation of solutions. Critical thinking cannot be obtained through rote learning ("button pushing") or spoon-feeding.

One of the major criticisms of self-paced material is the push-button nature of some available resources. Students are told to press keys to perform functions at a given time. When students are confronted with "real-world" situations, they do not know what to do or why to do it. Tutorials and other self-paced material may serve well as introductory lessons, but the test of students' abilities to adapt come

from advanced applications that require critical thinking. Teacher-developed learning activities can provide students with challenging learning environments.

Some possible teacher-developed learning activities for word processing may include these:

* Provide students with a resource for layouts of memos, letters, or reports. Give them the raw information needed to perform a task. The memo or letter may be partially handwritten or out of the "normal" order. These are situations they will encounter in the work world (see Figure 2, Sample Word Processing Problem).
* Make documentation available to students and encourage use of help screens so they can look up how to do something rather than just raising their hands to ask for help. Insist that they use those materials to learn to solve problems in procedures and/or formats.
* Develop building-block exercises where one exercise builds or develops from another. For example, generate a table for one exercise, and have the students merge the table into a report in a later exercise. Avoid, when possible, using files that are already keyed. If students indicate that they cannot key fast enough, assess the curriculum for a keyboarding prerequisite. When students constantly use pre-keyed files to try different word processing applications, they do not get a full range of experiences needed as reinforcement.
* Provide a table of proofreader marks, and give exercises using the marks. Help the students to develop proofreading skills, and give them exercises in thinking.
* After teaching a word processing application, have the students brainstorm on other uses of the application. Have them develop a list of other uses of the application. Columns, tables, footnotes, boilerplating, and mail merging are good applications for brainstorming activities.
* Develop exercises that do not quite fit on a page to teach widows and orphans protection, page numbering, or multi-page memos. Not everything is simple in the real world!
* Provide examples of inconsistencies of words in an example of search and replace applications. Use upper and lower case letters differently to show the exacting nature of most of these commands and the value of replacing with confirmation.
* Develop an exercise with two jobs active at one time to show how windowing works. Put a "flash/urgent" side bar or small job to be done within another job (advanced applications classes only).
* Have students develop a boilerplate lease, will, letter, or other document, and give the variable data on the side (see Figure 3, Sample Boilerplate Exercise).

Figure 2

Sample Word Processing Problem

Word Processing Assignment 3--The Rough Writer

You receive the following assignment on your desk. The letter is hand written on a napkin with 2 post it notes attached.

La Faux Pas
Dear Mr. Johnson:
I received the shipment of computer software today and found a problem with the order. It seems that the version of Acapella Toolbox was the wrong version. The version I received was the Windows version which only allows the user to prepare graphics. The correct version should have been the MULTIMEDIA version.
I realize that the MULTIMEDIA version comes as an upgrade. We will send unopened version to Jean Johns your sales rep at Chicago outlet and will cancel the purchase order number 67902 and issue a new order.
Sincerely,
Bill Waldron
Multimedia Specialist
Send this to
Mr. John Johnson
1522 Windy City Plaza
Chicago, IL 60603
Sorry . . his name is William Johnson
. . also spell out sales representative

Instructions:

Use the pages from the style manual section of your class notes to determine the layout for the letter. Check off each item as you complete them.

____ Prepare a letter in modified block style. Key the letter and save it as TOOL1.
____ Run the spell checker.
____ Proofread the letter and make any corrections needed.
____ Using confirmation, substitute the name JOHN with WILLIAM.
____ Save the file.
____ Print out the letter.

BEST COPY AVAILABLE

Figure 3

Sample Boilerplate Document

WordPerfect 6.1
Boilerplate Exercise

What it is:

Boiler plating is a means of providing stop codes in a document which allows a user to enter variable data into an existing file (primary file). Unlike merging where a primary file and secondary file exist, in this process the secondary file is replaced by input from the keyboard for each item. This application is suited for low-volume applications like leases, contracts, or letters where there are few copies. Use the mail merge feature if there are large volumes of letters or documents.

How it works:

1. Set up the primary (form) file.

During keying of the primary file (letter or document), the user enters a set of text and merge codes for each point in the document where information is to be entered from the keyboard. While creating the primary file, the user selects KEYBOARD from the Merge Feature Bar. The screen shows an Insert Merge Code prompt for KEYBOARD, and prompts the user to enter the type of prompt to be displayed when the file is merged. Save the primary file when completed. Clear the screen for the merge.

2. Merge.

With a cleared screen, begin the merge process by retrieving the primary (form) file and selecting Merge from the Tools menu. There will be no data file (secondary file). As the merge progresses, a Merge Message will appear at the bottom of the screen telling the user what information to enter. The merge pauses at the keyboard command.

Action: The Problem

Enter the following primary file:

Date (entered from system)

Enter 3-line address (Enter Name, Address, City State Zip)

Dear (Enter Name):

Our records indicate that you have not responded to our letter of (Enter date of letter). If you have not received the letter, please contact us soon. As a valued customer, you should not miss out on our special sale.

Sincerely,

Your name

Merge Data:

Mrs. William Warren
1210 Grove Street
Clarks Summit, PA 18415

July 15

Second letter:

Mr. Stanley Colinsky
103 Main Street
Dickson City, PA 18511

July 31

* Have students read an article about hardware or software and prepare a report that gives a summary of content, reaction to the content, and source citations. This activity could be used to reinforce learning of concepts such as footnotes or endnotes.
* Have students in an advanced applications class compose a story and place it into two columns with a graphic to emphasize the nature of the story (See Figure 4, Sample Two-Column Story).
* Assign a project to students for developing a personal letterhead using the macro features of the word processing package.
* Develop an exercise containing errors that spell checkers do not find.
* Have students develop a mail merge letter announcing an upcoming event, or use school club membership lists announcing a fund-raising event.
* Keep a real-life horror file of bad examples. Present the bad example, and have students correctly revise the application. Examples may include poorly produced letters, memos, reports, or flyers. Be sure to block out names to protect the parties involved. This application could also be used in desktop publishing.
* Have students compose or enter a multi-page document about their word processing package to show how to use headers and footers. A side benefit may be in the form of the conceptual knowledge gained from the document.
* If your software package utilizes a menu tree, have students draw and label the menu tree using the line draw features of the package.
* If your software uses icons for functions, develop a pencil-and-paper matching application. Give students a list of functions to be performed and pictures of icons for matching. This review may also be used for desktop publishing applications. Figure 5 illustrates the use of pictorial or icons and matching.

The learning curve for most desktop publishing courses requires extensive background information before students can jump into the application program. Information about fonts, kerning, leading, and design are not hands-on. Teacher-developed learning activities reinforce definitions, hardware and software needs, and design concepts for desktop publishing. Following are some examples of teacher-developed learning activities for desktop publishing classes without actual computer use:

* Have students research hardware and software needed for desktop publishing and report their findings. Give them a generic list such as laser printers, mice, digitizing pads, scanners, and secondary storage units as a starting point. Have students report on new products and provide information such as the cost range, manufacturer, compatibility with other equipment, and use in desktop publishing.

Figure 4

Sample Two-Column Story

A Man Goes to the Store

This is the story of a man who went to the store to get some things for his wife who was sick with the flu. He was to pick up some cold medicine and some fruit juice.

While on the way he came upon a little puppy that was black and white, or white and black depending on how you looked at the dog. The dog had a white mark over the left eye and walked with a limp due to some canine injury.

While at a light, the dog stepped off the curb and was hit by a big bus that had a sign that said "Express" in the window. The man saw that the dog was not injured and also noticed that the dog was no longer limping. To make a long story short, the bump cured the limp!

Continuing on his trip to the store, the man came upon a little kitty. The kitty was yellow. The kitty looked as if it had used up at least eight of its nine lives. Poor little kitty. The man picked up the cat and carried it with him to the store.

Finally, the man got to the store. While he looked for his shopping list, the kitty got away. The man went to the office of the store and had the cat paged. Naturally, the cat was at the fish counter partaking of the "fresh" salmon that comes in on every other third Tuesday. The cat came at once.

The man forgot what he went to the store to get and grabbed the cat and went home to his wife.

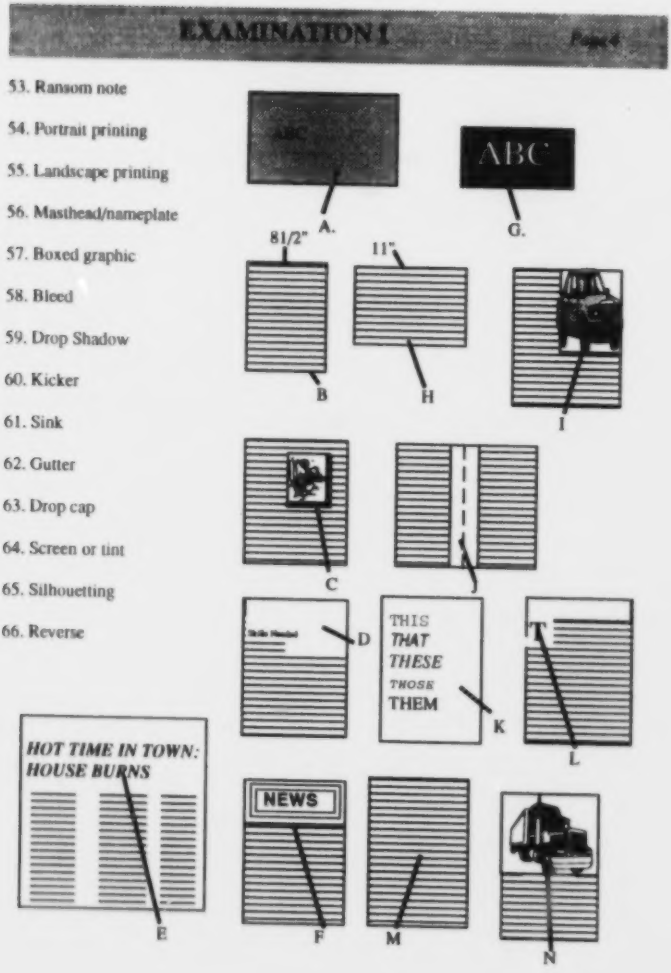
While on the way home, the man saw a bird sitting on a fence. The bird flew away.

Bingo! The man thought. That's why I was going to the store! My wife has the FLU. The man took the cat, went back to the store and finally went home. His wife was no longer ill by the time he got back.

BEST COPY AVAILABLE

Figure 5

Use of Icons or Graphics for Matching Exercise



This exercise can be completed with a word processing package or can be handwritten. The cumulative results should be an "ideal" configuration for desktop publishing that will provide students with a conceptual knowledge of what is needed for desktop publishing and cost ranges for desktop publishing systems. If a word processor is used, the file can be used for future exercises in importing files from a word processing package into a desktop publishing software package.

- Use portfolios to reinforce learning of desktop publishing terms and concepts. This exercise will require teachers to provide students with definitions and examples of the concepts by using a glossary or text. Teachers may provide the terms in a word processing file and have students import the text into the desktop program. Assign students the task of using old magazines or newspapers to find examples of commonly used desktop publishing tools and techniques such as dropped caps, pulled quotes, tint screens, dingbats, dropped shadows, reverses, sinks, kickers, bleeds, ears, mastheads, bylines, grid and axis layouts, and marginal side headings. Students should cut out the example, paste it on a sheet of paper, label it, and be prepared to share it with the class (see Figure 6, Sample Portfolio). The time spent going over the examples in class exposes the class to a wide variety of techniques.
- Use similar portfolio guidelines, and have students find examples of different types of fonts, font usage, type styles such as serif and sans serif fonts, examples of exaggerated leading or kerning, use of a wrong font for the theme, and customized fonts.
- Develop a portfolio that includes desktop publishing "no-nos" such as ransom note design (using too many fonts in one layout), columns too wide or too narrow, tombstones, or poor page layout. For this portfolio students should have examples ahead of time (finding examples is a long-term task for the instructor).
- Require alternatives to portfolios that may include "show and tell" or "flash cards." Have individual students find examples and show them to the class; or collect examples and use an overhead to have students identify the items.
- Have students brainstorm on applications that can use desktop publishing in their areas of study. Have them list possible uses of desktop publishing for their fields.

When working with the desktop publishing program itself, sequence the learning activities according to difficulty of the application. Most problems with desktop software are in text applications rather than the graphics. It is reasonable to provide many activities that deal with text handling such as breaking, stretching, and combining text blocks, using the on-board text editor, checking spelling, and flowing text from column to column or page to page. Many of the

Figure 6

Sample Portfolio for Desktop Publishing Terms

LEARNING ACTIVITY 4

DESIGN PORTFOLIO: TYPE STYLES AND TOOLS OF EMPHASIS

Purpose: The purpose of this activity is to acquaint you with the different terms related to publishing.

Procedures: Using blank pages, scissors, and tape, find examples of the following. If you CANNOT find an item, give your own definition of the example:

Tombstone (hard to find)
Reverse type style
Outline type style
Drop cap or dropped cap
Shadow type style
Screen or tint screen
Pull quote
Kicker
Silhouetting
Bleed
Masthead
Sink
Rules (lines)
Jim-dash or cutoff rule
Dingbats
Sidebar
Gutter
Ears
Marginal subheads
Nameplate
Axis layout
Grid layout

Tips ricks and idbits

Cut and tape your examples on a sheet of paper and label the example.

To save paper, you may put several examples on each page. Please keep The paper size to 8.5 x 11 inches.

hands-on learning activities provided by publishers require all students to use the same text, graphics, and layouts for completing projects. Desktop publishing is a combination of art form and technology. Since students are from divergent backgrounds, allow them some artistic freedom; however, provide guidelines. Instructions for each hands-on activity should specify its purpose, guidelines, and necessary tips or concepts for the task.

During the introductory stages, students should learn the basics of the menu system, toolbox functions, default settings and the screen layout. Assuming there has been a conceptual background on fonts and features of desktop publishing, lessons can be arranged with the following teacher-developed learning activities:

* Start with a tour of the program activity. Most desktop programs allow the user to change defaults when a file is NOT opened. Provide an exercise for students with a modified version of a "pilot's pre-flight checklist." The check list may include the type specifications as 12 point Times Roman with automatic leading and normal kerning. Tell students to read the menus and check whether the defaults are what appear on the checklist. The tour will reinforce the terms learned and provide students with a feel for the program and its menus. Proof of completion of the activity may include brief handwritten answers to questions related to defaults shown on the screen.
* Assign a one-page flyer as an early production activity. The purpose of the activity should be to introduce students to the toolbox and working with text, boxes, circles, and lines. The flyer may announce a meeting, advertise a product, or be instructional (such as a warning sign). Since students may be from many disciplines, let their creativity flow. At this point the teacher may ask students not to use clip art or other graphics. The purpose is to use the existing toolbox features (See Figure 7, Sample DTP Exercise).
* Introduce the cut-and-paste feature of DTP software by assigning an activity of creating business cards or tickets. If the student can make one good model and knows how to cut and paste, several cards or tickets can be made on the same page. The use of column guides and rulers is also required for this exercise to place the items correctly on the page.
* Introduce the use of master pages by having students develop certificates. The certificates can be developed on the master page, and subsequent pages can be used to fill in required data on the certificates. In this exercise, more than one certificate should be printed to indicate that the certificate is correctly placed on the master page.
* Illustrate and reinforce the concept of file importing by having students develop a two-column, one-page layout using text from their hardware or software report done on a word processor. The output can be enhanced by adding a graphic, a pulled quote, a dropped cap, or all three.

Figure 7

DTP Exercise

LEARNING ACTIVITY

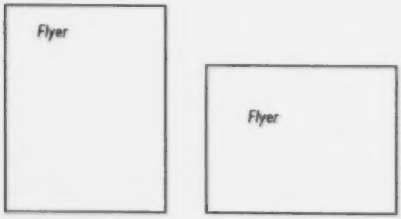
TOOLBOX: ONE-PAGE FLYER

Purpose:

The purpose of this activity is to acquaint you with text and toolbox functions of PageMaker.

Procedures:

Prepare a one-page flyer on any subject area of your choice. The flyer can be an advertisement, meeting announcement, or any application you choose. Be sure that the flyer illustrates the use of text and toolbox (boxes, circles, etc.) Enter the text using the default sizes and use the text highlight technique to change styles, alignment, and sizes. You may use either portrait or landscape layout (tall or wide).



Tips, Tricks & Tidbits

Before opening a new file, go through the PageMaker menus to determine the defaults for the file. REMEMBER--whatever defaults are when you open a file, they will stay with the file forever and ever and ever and ever! Check them out with files closed first!

Stay away from the **space bar** to align text. Due to kerning, PageMaker output is unpredictable when spaces are used.

- Assign an exercise or activity using the scanner and the same text file from the hardware or software report. The exercise may also show text wrapping and/or graphic sizing or cropping.
- Develop activities to create a two- and/or three-column flyer with information on the front, back and inside to teach students how to use master pages to set up complex publications. Text and graphics rotation techniques can be explored in the form of return addresses or side layouts.
- A business form make-over or customizing activity incorporates text, lines, and graphics (company logos). This activity may involve the student in scanning, clip art, or other draw or paint programs as well as learning about import filters (files used to convert software into the format used by the DTP package).
- A culminating activity (perhaps a final exam project) asks students to develop a newsletter or magazine about desktop publishing. Give the students about five article titles, and have them compose the articles and put them together in a multi-page format, including graphics. Grading would include both content and technique used. Figure 8 illustrates a take-home or "carry out" examination project.

Simulations

Simulations take many forms. Simulations that show how to perform tasks using hypermedia (continue buttons, branching, etc.) are usually found on laser discs and other multimedia systems. Comsell training programs for Windows and WordPerfect for Windows use laser disks that cost over two thousand dollars. These systems are expensive at present and would be suitable for small groups using individualized instruction. Another more common and less expensive form of simulation is a printed practice set.

Printed simulations for word processing are best suited for advanced classes that emphasize increased speed of production. Materials with emphasis on word processing for certain professions, such as medical or legal office environments, are available from publishers. Several packages now available provide realistic activities by incorporating terminology, forms, and practices of the professions. For introductory courses, use more generic applications because of the divergent population of the classes. Memos, letters of application, and reports are examples of more generic applications.

Printers cause another problem with packaged simulations available for classroom use. Many packages have pre-printed letterheads and forms. If the printers do not allow for single sheet feeding, using the forms is impossible. Laser printers may not be available or may be networked, posing time problems for students. Students will not know when their jobs are going to print, and they may

Figure 8

Take-Home or "Carry-Out" Exam Project

Carry Out Final Examination

Exam Guidelines:

The purpose of this examination is to give you the opportunity to integrate your **knowledge** of desktop publishing and display your **skills** in the use of PageMaker and other software tools used in the field. The guidelines are simple. The essays or articles should be written in a word processing package and **PLACED** into a multi-page document which illustrates your creativity and knowledge of PageMaker.

Each article should be about a page long (12 point type size). Use titles, pull quotes, graphics or any other tool that will make your document interesting. The examination will be due on the last day of class and will be graded on content and technique (50-50 basis). Each "article" will be worth 10 points. 10 more points will be used for overall design and layout for a 60 point total for the exam. Be sure to save your file(s) as often as possible. When printing, try to print one page at a time if you have problems. Since exams will NOT be returned, make **2 copies** of the final printout if you want it for a portfolio.

Topics:

1. "Desk Top Publishing: Ways to Stay Up-to-date in a Changing Field and Personal Reactions from the Editor" *(you)*

2. "Dream Machine: Hardware and Software for an Ideal DTP System"

3. "Questions from Readers and Answers from the Expert (you)"
   Why do I get different fonts on different machines I use?
   What are the ups and downs of using a network?
   What kind of printer do I need when I have a 1200 DPI scanner?
   Why do I have problems importing files from my word processor?

4. "20 Possible Uses of PageMaker in _____*(your major or occupation)*"

5. "15 Great Techniques to provide emphasis in PageMaker"

BEST COPY AVAILABLE

get their output on other students' forms. Coordinate packages with printer configurations in your classroom.

TESTING AND EVALUATION FOR WORD PROCESSING AND DESKTOP PUBLISHING

Probably the most difficult task of a teacher is evaluation of student performance. Questions about reliability and validity of a grade, issues of grade inflation, concerns about performance in later courses, and retention of subject matter are warranted and are an expected part of teaching.

Evaluation should be correlated to the objectives of the course and be measured from a variety of sources. Testing is only one form of evaluation. As we understand more about learning styles, the overall evaluation must measure using a variety of student activities.

Evaluation and Curriculum

In a curriculum that relies on prerequisite courses and performance, evaluation issues transcend individual courses, have a lasting impact on the total curriculum, and point more and more to students taking responsibility for their own learning. Students may lose their word processing or desktop publishing skills because of a lack of use over time. They may also need instruction to use a new software version. They must be ready to relearn and to update their skills to new software versions whenever needed.

What can you do to help yourself and your students overcome the problem? Provide for use of word processing in future courses, provide remedial or brush-up exercises for classes that will use software, and most importantly, provide students with learning tools to maintain their skills, learn new skills, and be more adaptable to change. Our major concern should be with what we do and how we evaluate when we have the students in our class. What can we do?

* Don't keep the objectives or content of the course a "secret." Make lists of tasks to be performed, and give them to the students so they know what they are supposed to learn and can take responsibility for learning.
* Provide a foundation of concepts rather than "button pushing." The buttons will change; the concepts will not.
* Use exercises and assignments that require students to THINK about what they are doing, USE resources available such as program documentation and books, and ACT accordingly.
* Allow students to use documentation during performance tests--they will have documentation on the job or in other classes. You will not be there to respond to a raised hand!

* Don't require rote memorization of keystrokes. Let students use templates if they need them. They will have them on their keyboards in the "real world."
* Encourage experimentation. If a student is on a different track that may attain the desired results, give suggestions, not criticism or reprimand. The "best" way today may not be the best way next week!
* Encourage creativity, particularly in desktop publishing activities. Using stock output expectations for everyone stifles experimentation and motivation.

Variety, Weights, and Frequency in Evaluation Mix

As in selection of instructional strategies, variety is an important consideration for evaluation. As with learning styles, students have different testing styles as well as different test anxiety levels. Providing a variety or mix of evaluation components helps to "level the playing field" and provide a more accurate assessment of individual performance or achievement. The mix may include objective tests, performance tests, exercises, reports, learning activities, and tutorials.

Other evaluation factors include ease of grading, concern for possible "borrowing" of other students' files or document printouts, amount of time on task, and time needed to provide students with feedback. Grading can be made easier by using key words, figures, or design layouts in homework or in-class exercises. Exercises that are done without teacher supervision can carry less weight in the total grade to prevent "borrowed" work, or projects can use the student's name within the document. Work that requires long periods of time should be a larger factor in grades than "quickie" assignments.

Frequency of evaluation is important to classroom learning and retention. Dempster's (1991) article on research about reviews and tests suggests that the more frequently testing is performed, the more learning takes place. Implications for word processing and desktop publishing suggest more use of tests and quizzes at spaced intervals rather than a few tests as summarizing or culminating activities. In addition, frequent, spaced practice is important to learning and retention.

Theory Tests

Theory tests for word processing and desktop publishing are among the most difficult to construct if they are done correctly. For word processing, these tests generally are used to evaluate lower levels in a taxonomy of learning. Definitions and concepts can be evaluated with theory tests. The downside to most theory test items is in their construction. Too often a test includes rote items that apply only in context, such as function key strokes and word processing tasks that are unique

to a given program. To evaluate higher-level learning, carefully construct and/or select item types (multiple choice, true/false, matching, and fill ins) for written tests. Use the criterion of importance: Does the item cover a broad, important concept, not just rote learning or package-specific concepts?

Two major considerations for theory tests are reliability and validity. Reliability measures the consistency of measurement. Does the test measure the same time after time, or do students score consistently on the first and second halves of the test or on even/odd items? Validity measures how well a test measures what it is supposed to measure. Does the test cover the objectives of the course and the content? Does it cover important concepts fairly and evenly? Unfortunately, a test can very reliably measure invalid subject matter; therefore, be careful to establish content validity of your tests. To help in this task, use a content list or grid first before developing or selecting test items.

Grids should list the content areas and indicate the level of understanding required for the content items. This list of terms, concepts, or tasks should be a vital part of the course anyway. Students should have the list so they know what they are supposed to learn. Then write or select test items to meet these learning goals. For example, if a goal is for students to understand how hidden codes work, the following test item for WordPerfect 5.1 would be inadequate:

Which of the following allows the user to see what is hidden in the file:
A. F3
B. Alt/F3
C. Shift/F3
D. CTRL/F3

A higher-level item might be:

Upon revealing codes, John finds the following:
[UND][und][BOLD]ACME SUPPLY[bold]
What will print?
A. ACME SUPPLY
B. **ACME SUPPLY**
C. <u>ACME SUPPLY</u>
D. **<u>ACME SUPPLY</u>**

The student in this example will have to know that files are stored sequentially in WordPerfect, indirectly that superfluous codes may be recorded by accident, and that the printer plays out a file in sequential order--a much higher level of understanding.

In small clusters of choices, matching items can be used. Avoid large lists that cover two columns of a page; they teach reading rather than word processing. In good matching tests, students quickly discount items that do not "belong" in a list as viable answers.

True/false items may give the student a fifty-fifty chance, which are not bad odds if the student does not know the material. Writing good true/false items that discriminate among levels of learning is difficult. Fill-ins are easy to construct but difficult to grade. These types of questions work well for quizzes or reviews or if a specific term is important for students to know.

Another variation of matching questions involves using icons or graphics. This concept works well to check whether students know how to use the toolbox in PageMaker, for example, or to evaluate knowledge of more icon-oriented word processing packages. Give a list of tasks to be performed and ask students to select the correct tool from the toolbox. The list of tasks should include more than one use of the tool. Figure 9 illustrates this application. When testing design terms, use thumbnails as examples and have students match the graphic to the desktop publishing term. Figure 10 illustrates this possibility.

A last caution about theory tests involves the use of published test banks. Test banks may have been generated by persons other than the text author, with no reported statistics on reliability or validity. They may not have been tested for validity and reliability. The items may not have been evaluated for correctness out of context. They may be correct for the situations included in the text, but in a broader application, other answers may be correct. You must be the judge of appropriate items from test banks to evaluate your students fairly and adequately.

With scanning equipment and test analysis programs, you can collect data on individual test item performance. Item difficulty and discrimination indices give better information on student performance. If you return a test to students to keep, your opportunity to create a good test bank diminishes. Give back the results on tests, but keep the test papers. As you evaluate the effectiveness of individual test items, you can retain, discard, or revise individual items and build a better test bank for future use.

Performance Tests

Hands-on performance tests are good measures of student performance in word processing and desktop publishing classes. To measure competency, set a time limit for completion of the examination problems. Provide several problems over one major unit. You can then give partial credit for uncompleted problems that are correctly formatted. Using a test diskette rather than printouts allows you to check a variety of learning, such as format, the use of special features, and hidden codes, much better than in printed form.

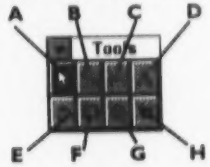
Figure 9

Sample of Tool Box Matching

Please do NOT write on the test booklet

Toolbox Matching:

*Use the following diagram to answer the next set of test items. Match the **BEST** toolbox icon with the listed tasks to be performed:*



33. Delete an entire block of text.

34. Stretch a graphic.

35. Crop a graphic.

36. Move a block of text.

37. Draw an oval.

38. Draw a line from the bottom left corner of a page to the upper right corner.

39. Change spelling of a word in text.

40. Make a vertical rule.

41. Rotate a square

42. Stretch an existing rectangle

43. Change the style of text for one word.

44. Add a word to text.

45. Draw a rounded corner shaded box.

46. Delete a circle.

47. Manually flow text on document.

Figure 10

Sample of Graphic Matching

Examination II

Picture Matching:

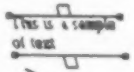
Match the BEST picture with the definition or description to the left.

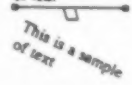
48. Rotated text
49. Skewed text
50. More UNplaced text in block
51. More placed text in block
52. "Complete" text block

A. s a m p l e

B.

C.

D. 

E. 

F.

As in other forms of evaluation, take care to assure the content validity of the performance test. Giving a test that does not adequately cover what is outlined in the course makes a performance test invalid.

Grading performance tests is time consuming; however, shortcuts can make grading easier. These shortcuts include checking key words or figures in problems, using a checklist of what should be included in the problem, and grading one problem for all students at a time before moving to another problem. These methods make grading more consistent and fair.

Performance tests for desktop publishing classes present another set of problems. Since the software and hardware are more sophisticated or most applications utilize a local area network, printing may cause problems. A laser printer may take up to a half hour to print a complex document with graphics. If this is the case, use a major project, completed as a long-term assignment, or other form of performance.

Other Evaluation Techniques

Individual or group projects are good evaluation tools, particularly in desktop publishing classes. State the scope of a project and the format in which the final product will be presented. Group projects need more attention to sharing the work load and more detail about how grades will be determined. Because they involve longer times and more synthesis of all students have learned, large projects require good organization, guidelines, and clarification.

A good project for small-group work in desktop publishing classes is working with graphics packages that can be imported into desktop publishing software. Have two or three students work with a draw or paint program and demonstrate to the class how it works. If there are several draw or paint programs available, they will get exposure to many programs and learn more about one particular program. Be sure to establish guidelines or formats for group reports so there will be consistency of the reports or presentations. Figure 11 illustrates some guidelines for a group project.

Projects encourage students to synthesize all they have learned. They must select among the tools and processes in the package. They can be creative. The quality of their final products shows what they know. Presenting their projects to other class members develops their skills in presentation, a valuable skill in the work place.

SUMMARY

The major role of the teacher of word processing and desktop publishing is to provide an environment conducive to learning. Characteristics of this environment should involve learners through a variety of hands-on activities.

Figure 11

Sample Group Project Guidelines

Group Members
(no more than 3 students):

_____

_____

_____

Package Assigned: _____

Date of Presentation: _____

Purpose:

The purpose of the group project is to explore the possible uses of various draw and paint programs with PageMaker.

Background:

Several draw and paint programs are available in the Lab. Most have similar functions and can be imported into PageMaker. In order for everyone in the class to become acquainted with several programs, each group will be assigned a program for exploration and will report to the class. Programs will be assigned by drawing.

Guidelines:

1. Elect a team leader to report the group's progress and to serve as the coordinator of the project.

2. Use tutorials, documentation, or on-line help systems to explore the program and be able to perform basic functions.

3. Using the basic functions of the package, prepare a graphics file and save it on your disk. If the package has several options for saving files, save the file in ALL formats available. Experiment with placing the file in PageMaker--find out what outputs will import into PageMaker.

4. Prepare a one-page report that summarizes the package features, gives the user reaction to how it works, and how it imports into PageMaker.

5. Prepare a brief oral report (about 10 minutes) that gives fellow students an overview and demonstration of the program. Be sure to include ALL members of the group in the presentation. Allow for questions from your fellow students.

Grading Criteria:

Grading of the group project will consider the quality of the written report (10 points) and of the presentation (10 points). All members of the group will receive the same number of points. It is up to the group to fairly distribute the efforts of the project.

BEST COPY AVAILABLE

Activities should be designed to allow students to take responsibility for learning by letting them know what the objectives are for the activity.

Since students learn differently, the variety of activities helps to provide an even playing field for the wide variety of student interests and abilities common to word processing and desktop publishing classes. The variety should also extend to evaluation methods from theory testing to hands-on performance tests.

Experimenting with different learning environments makes teaching and learning more interesting and challenging for teachers and students. We cannot "teach" anything unless students want to learn. Anything that can be done to encourage learning should be tried.

WORD PROCESSING AND DESKTOP PUBLISHNG REFERENCES

Dempster, F. N., (April, 1991). Synthesis of research on reviews and tests. Educational Leadership, 48(7), 71-76.

Dunn, R. (March, 1981). Teaching in a purple fog: What we don't know about learning styles. NASSP Bulletin, 65(443), 33-36.

Dunn, R., Baudry, J. S., and Klavas, A. (March 1989). Survey of research on learning styles, Educational Leadership, 46 (6), 50-58.

Everett, D. R. and Lund O'Neil, S. (Spring 1990). Competencies for information systems workers. The Delta Pi Epsilon Journal, 32(2), 41-55.

Hunter, J. (January 1992). Teaching higher-level thinking skills. Instructional Strategies: An Applied Research Series, Delta Pi Epsilon, 8(1), 1-4.

Nellermoe, D. A. (May/June 1993). Problem solving in software tool courses: Beyond the basics. Journal of Education for Business, 68(5), 273-275.

Newell, J. (1984). Advance organizers: Their construction and use in instructional design. Instructional Development: The State of the Art, 1, 262-274.

Orr, C. L. (Spring 1990). A comparison of achievement and attitudes of postsecondary students taught by two methods of instruction in word processing. The Delta Pi Epsilon Journal, 32(2), 71-85.

Pollard, C. and Kizzier, D. (Spring 1992). A comparison of technologically mediated instructional systems: Interactive videodisc and videotape, The Delta Pi Epsilon Journal, 34(2), 83-95.

Reiff, J. C. (1992). Learning styles: What research says to the teacher series. National Education Association Monograph, 42 pages.

Yelon, S. (November 1985). Making decisions about pretesting: It's not a simple matter. Performance and Instruction Journal, 9-10, 13.